SCALING ANALYTICS WITH PANGEO: XARRAY + DASK + ZARR

RYAN ABERNATHEY

CEOS WGISS-49

WHO AM I?

Physical Oceanographer

Ph.D. From MIT, 2012

Associate Prof. at Columbia / LDEO

https://ocean-transport.github.io/

Core developer of Zarr

Core developer of Xarray

Co-founder of Pangeo

Open Source Advocate

WHAT IS PANGEO?

“A community platform for Big Data geoscience”

- Open Community
- Open Source Software
- Open Source Infrastructure

PANGEO COMMUNITY

HTTP://PANGEO.IO

Lamont-Doherty Earth Observatory
COLUMBIA UNIVERSITY | EARTH INSTITUTE

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Met Office

USGS

ECMWF

GEOMAR

UNIVERSITY of WASHINGTON
eScience Institute

BIDS
BERKELEY INSTITUTE FOR DATA SCIENCE

jupyter

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

CSIRO

ANACONDA
Powered by Continuum Analytics

JUPITER™

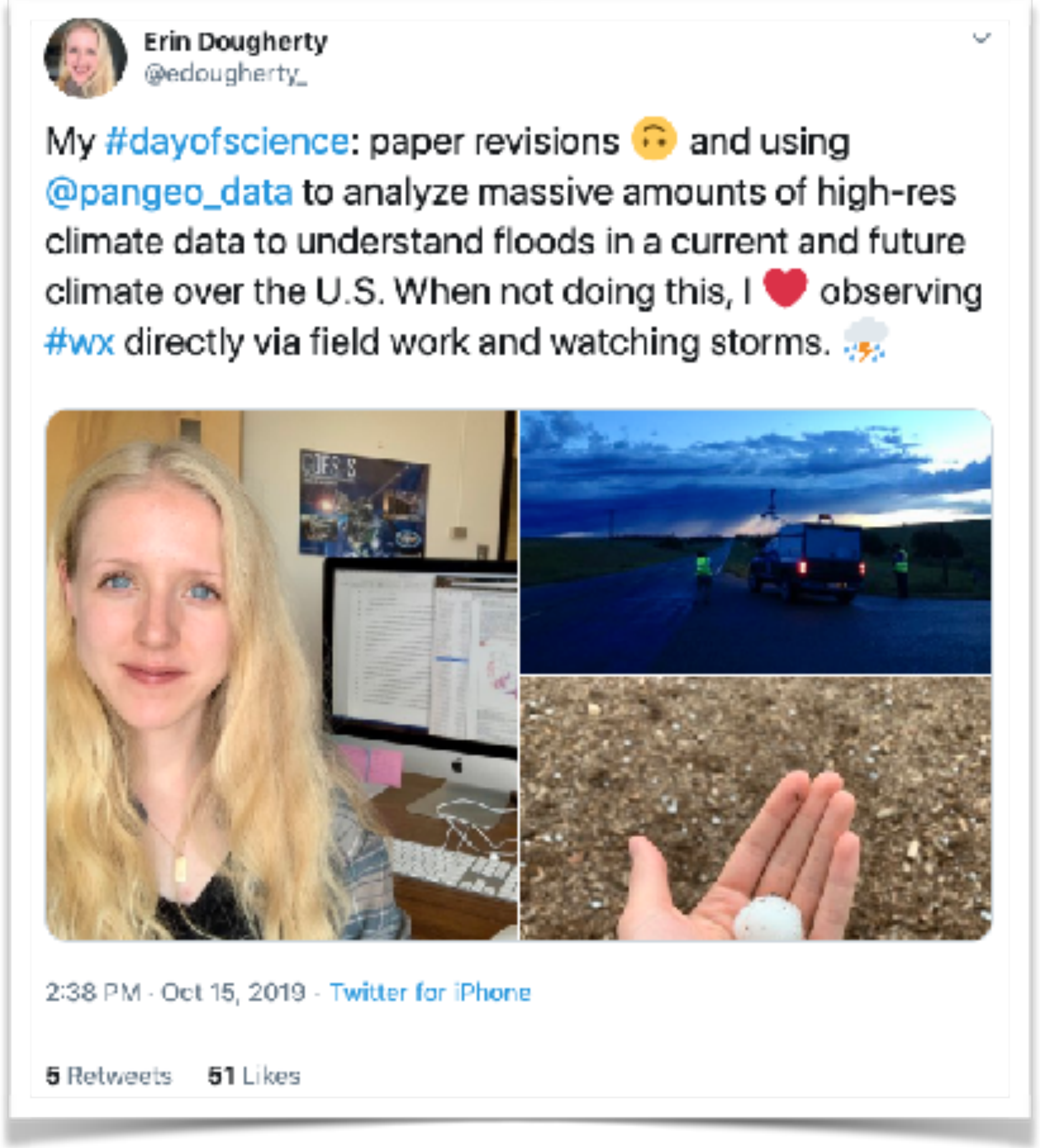
element84

developmentSEED

Rh RHODIUM GROUP

CLIMACELL
Weather Revealed

GRASS-ROOTS ADOPTION

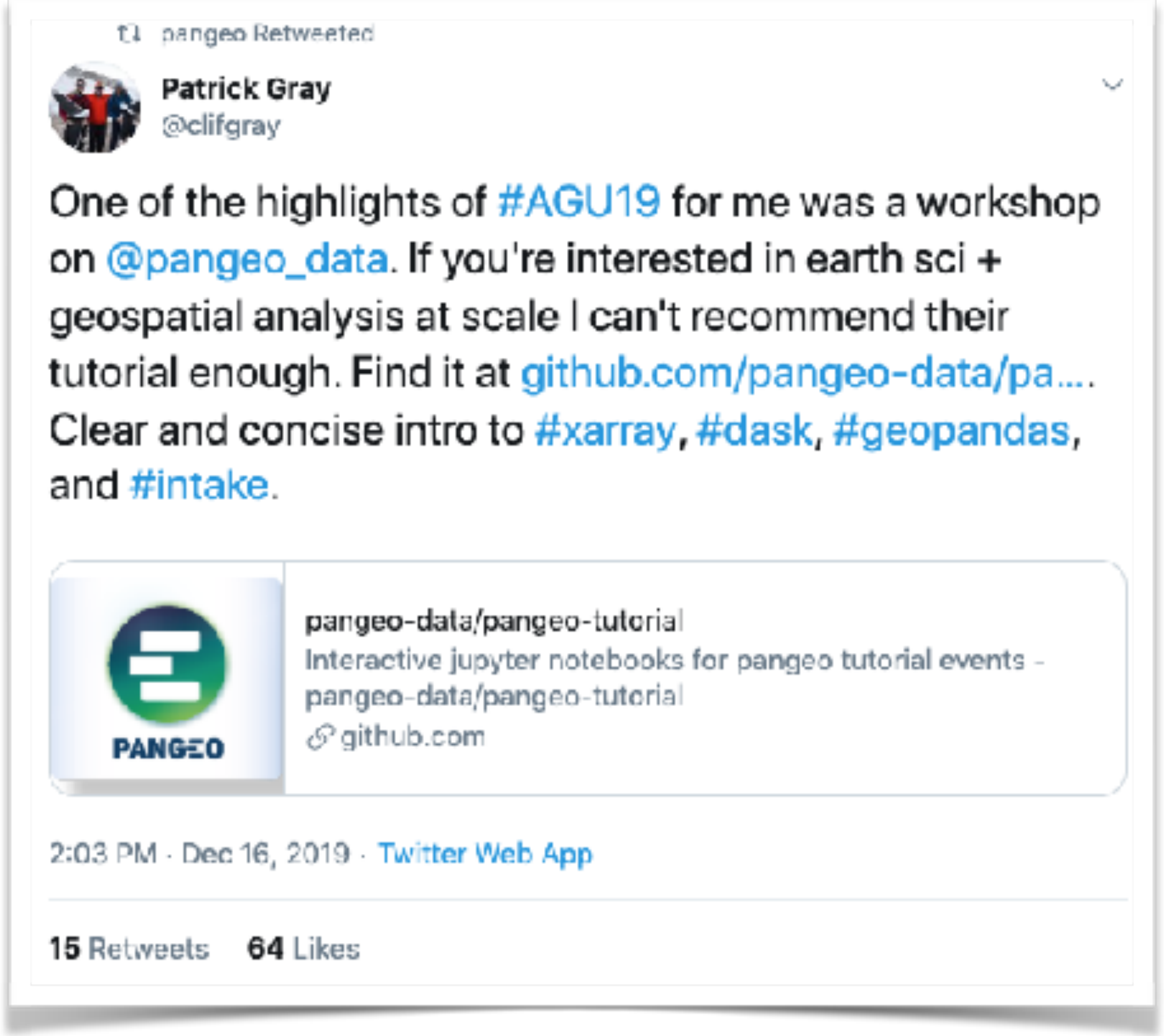


Erin Dougherty
@edougherty_

My #dayofscience: paper revisions 🙃 and using @pangeo_data to analyze massive amounts of high-res climate data to understand floods in a current and future climate over the U.S. When not doing this, I ❤️ observing #wx directly via field work and watching storms. ⛈️

2:38 PM · Oct 15, 2019 · Twitter for iPhone

5 Retweets 51 Likes



pangeo Retweeted

Patrick Gray
@clifgray

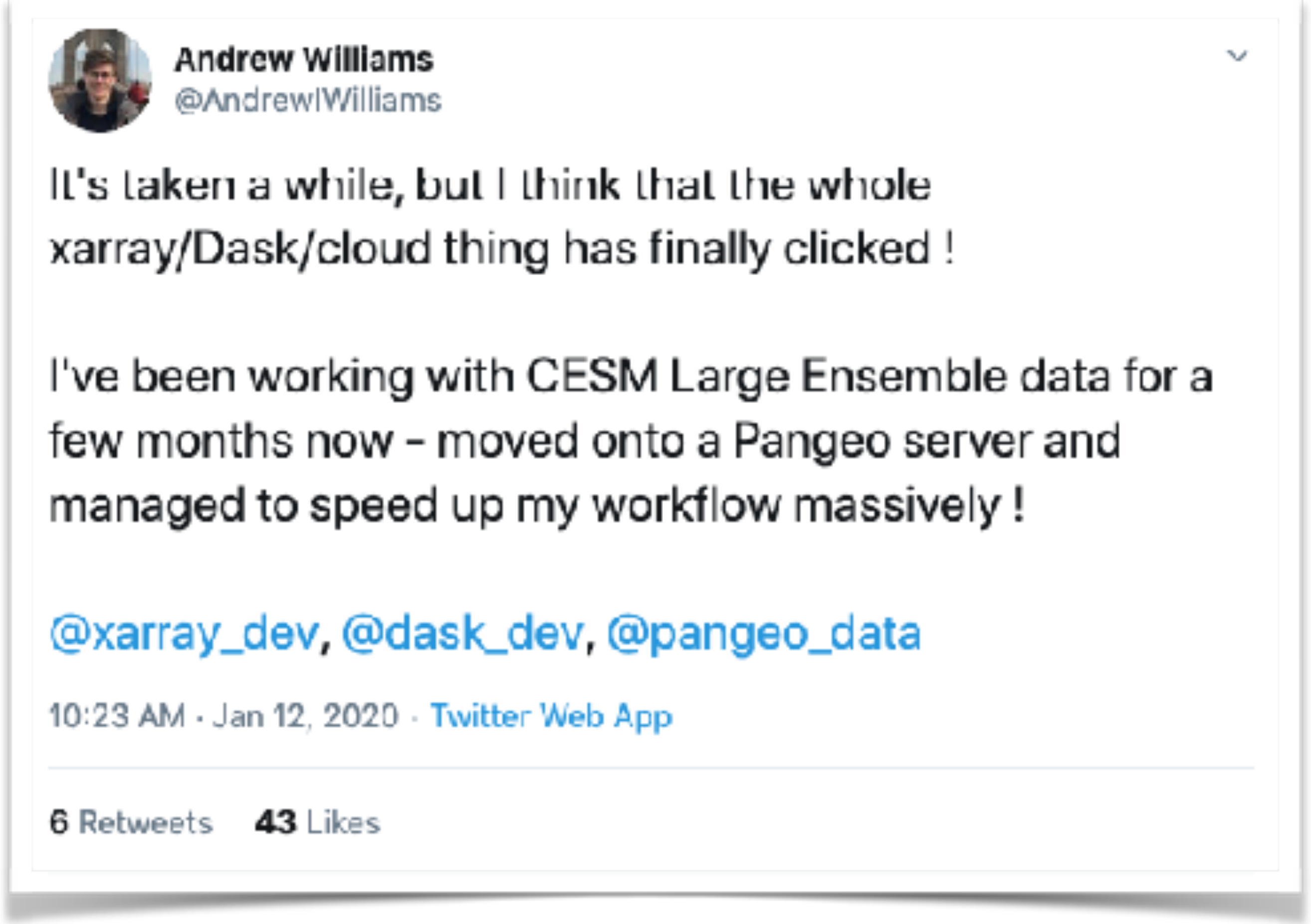
One of the highlights of #AGU19 for me was a workshop on @pangeo_data. If you're interested in earth sci + geospatial analysis at scale I can't recommend their tutorial enough. Find it at github.com/pangeo-data/pa…. Clear and concise intro to #xarray, #dask, #geopandas, and #intake.

PANGEO

pangeo-data/pangeo-tutorial
Interactive jupyter notebooks for pangeo tutorial events - pangeo-data/pangeo-tutorial
github.com

2:03 PM · Dec 16, 2019 · Twitter Web App

15 Retweets 64 Likes



Andrew Williams
@AndrewIWilliams

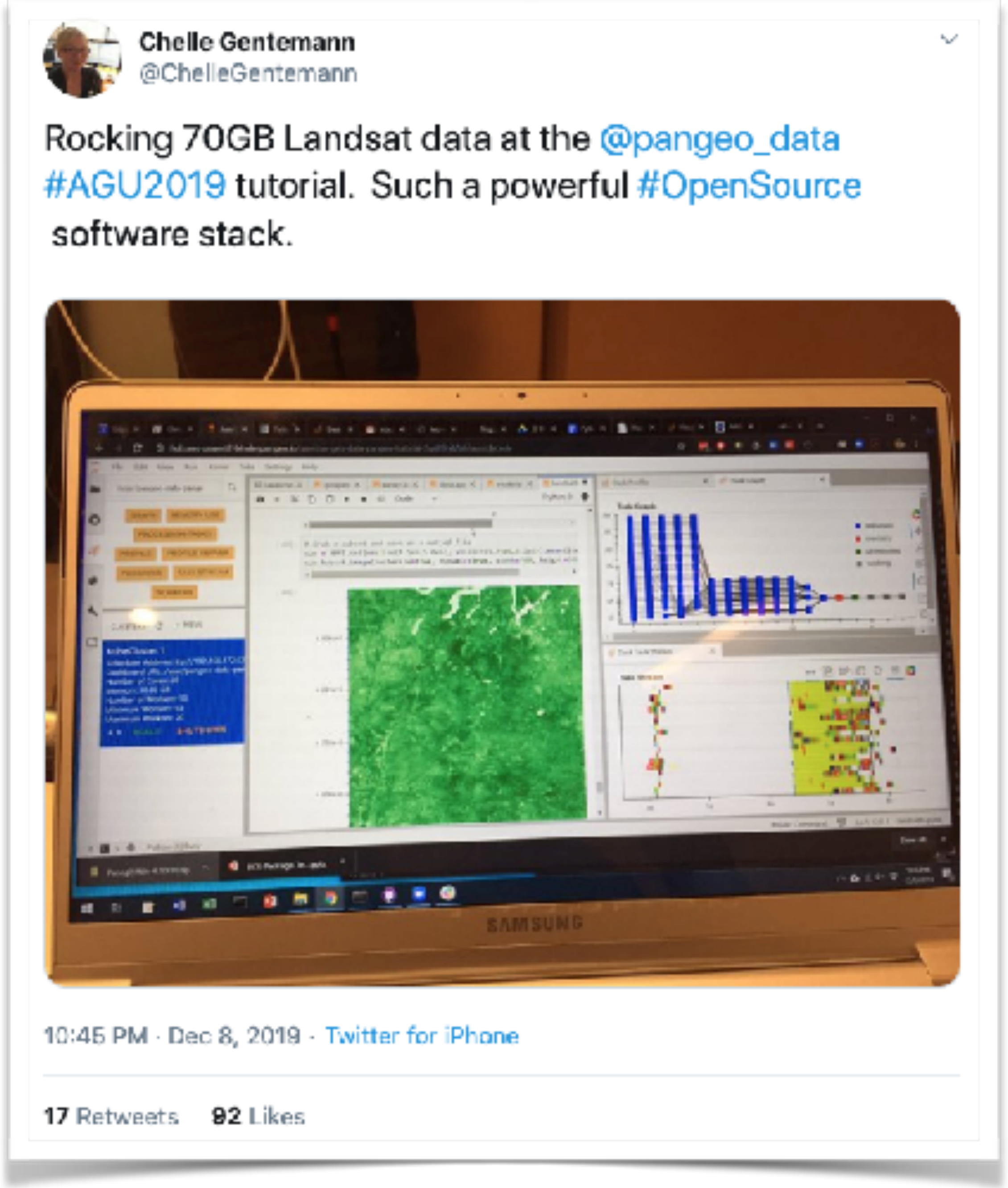
It's taken a while, but I think that the whole xarray/Dask/cloud thing has finally clicked !

I've been working with CESM Large Ensemble data for a few months now - moved onto a Pangeo server and managed to speed up my workflow massively !

@xarray_dev, @dask_dev, @pangeo_data

10:23 AM · Jan 12, 2020 · Twitter Web App

6 Retweets 43 Likes

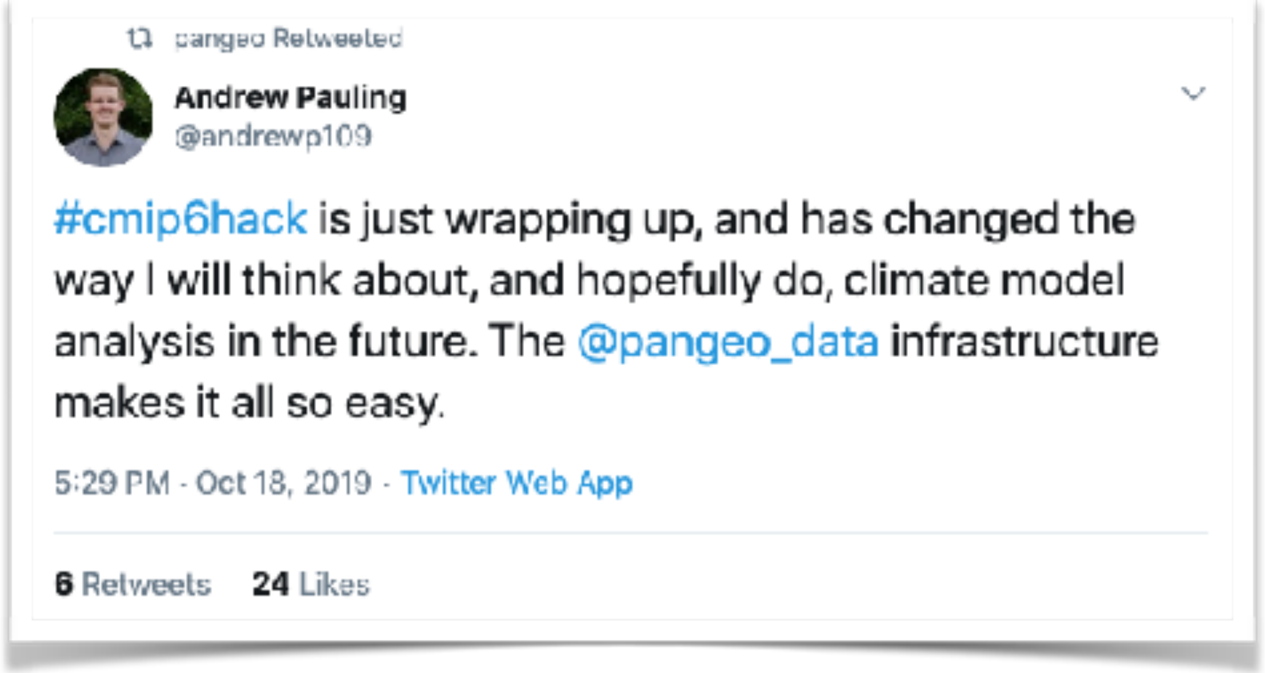


Chelle Gentemann
@ChelleGentemann

Rocking 70GB Landsat data at the @pangeo_data #AGU2019 tutorial. Such a powerful #OpenSource software stack.

10:45 PM · Dec 8, 2019 · Twitter for iPhone

17 Retweets 92 Likes



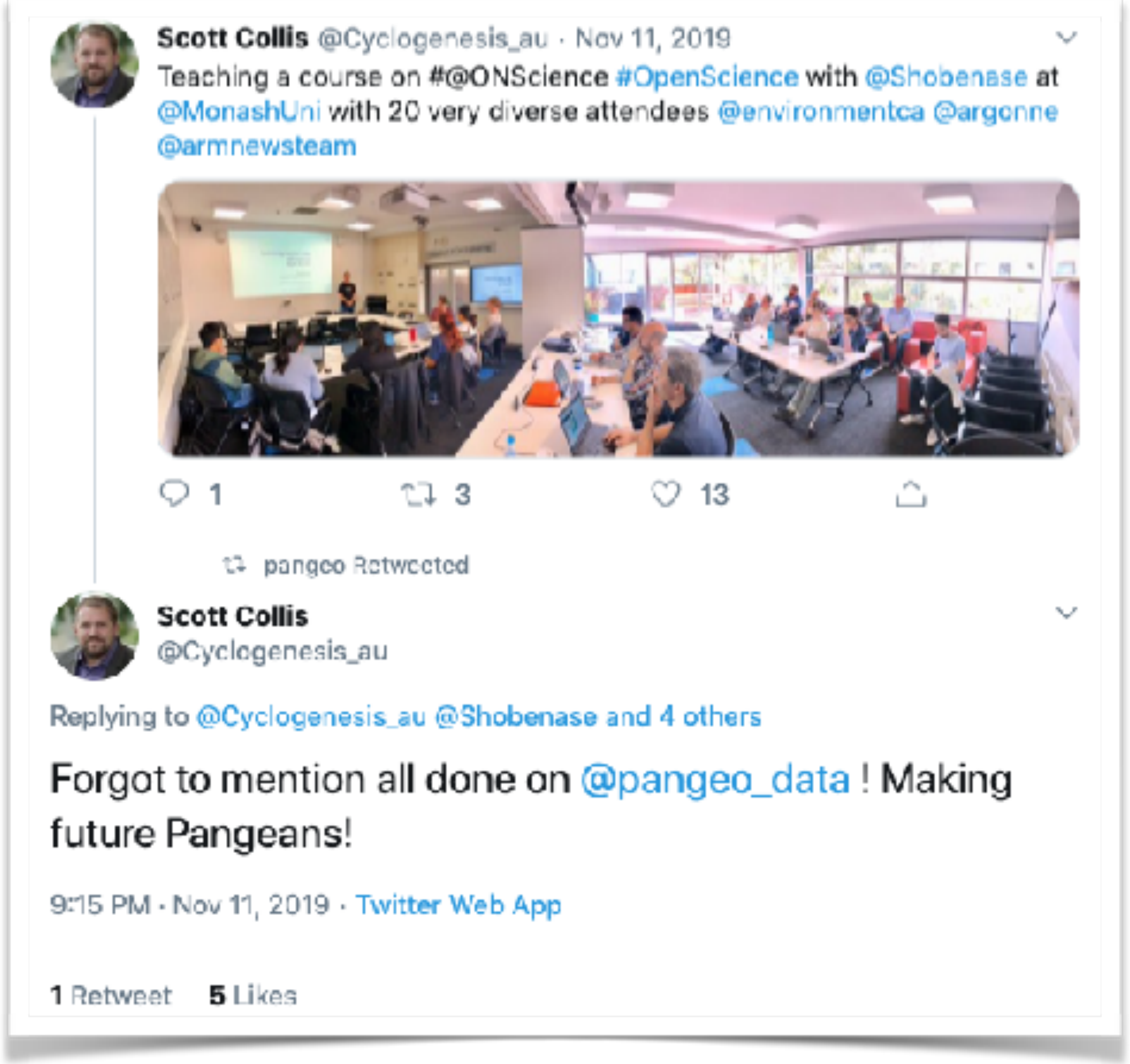
pangeo Retweeted

Andrew Pauling
@andrewp109

#cmip6hack is just wrapping up, and has changed the way I will think about, and hopefully do, climate model analysis in the future. The @pangeo_data infrastructure makes it all so easy.

5:29 PM · Oct 18, 2019 · Twitter Web App

6 Retweets 24 Likes



Scott Collis @Cyclogenesis_au · Nov 11, 2019

Teaching a course on #@ONScience #OpenScience with @Shobenase at @MonashUni with 20 very diverse attendees @environmentca @argonne @armnewsteam

1 3 13

pangeo Retweeted

Scott Collis
@Cyclogenesis_au

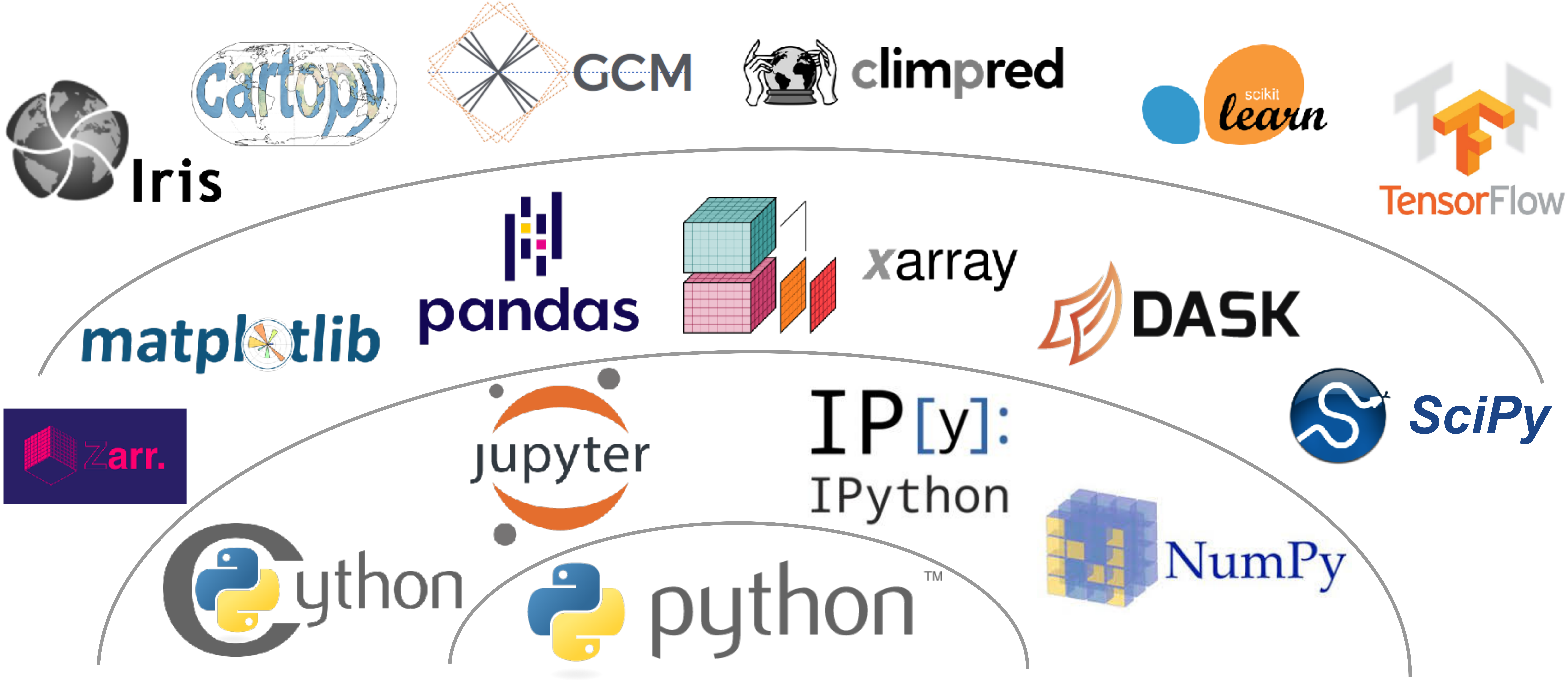
Replying to @Cyclogenesis_au @Shobenase and 4 others

Forgot to mention all done on @pangeo_data ! Making future Pangeans!

9:15 PM · Nov 11, 2019 · Twitter Web App

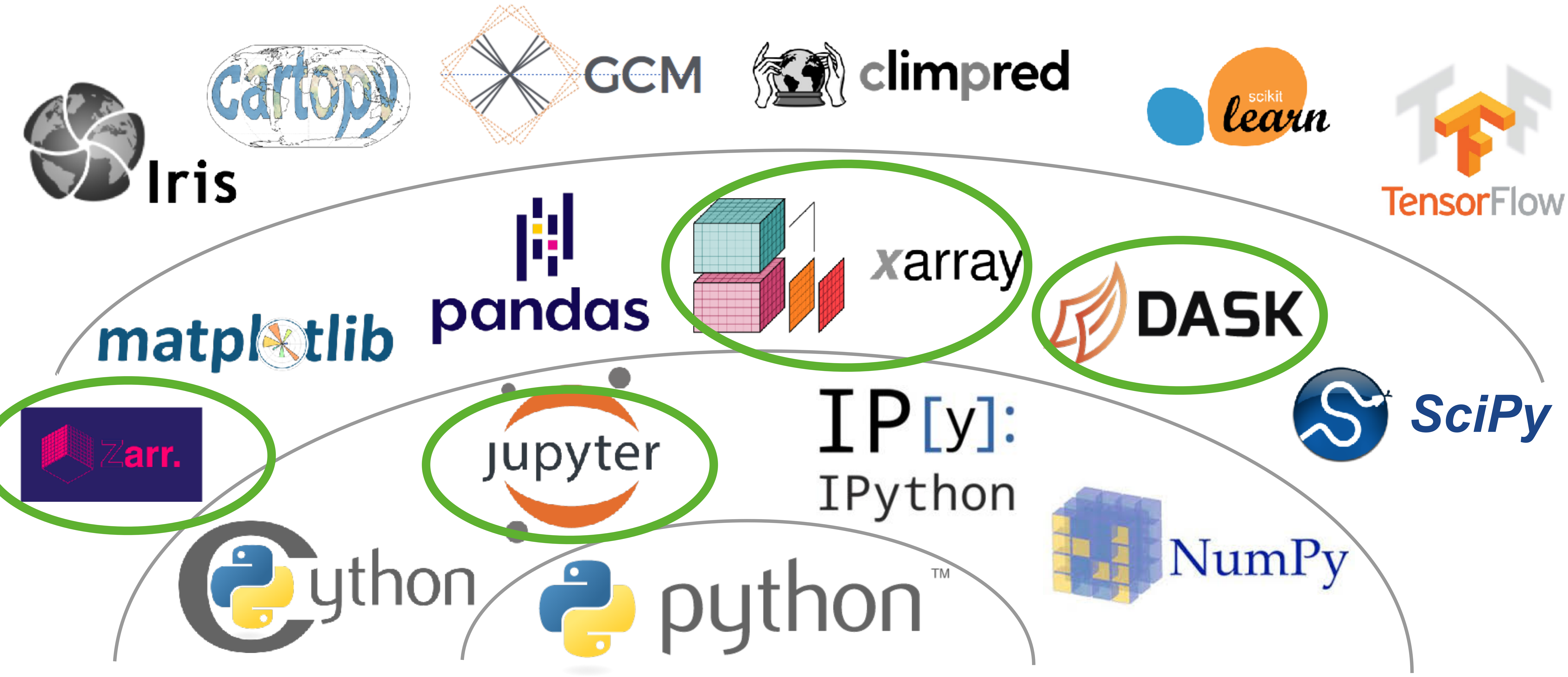
1 Retweet 5 Likes

PANGEO SOFTWARE ECOSYSTEM



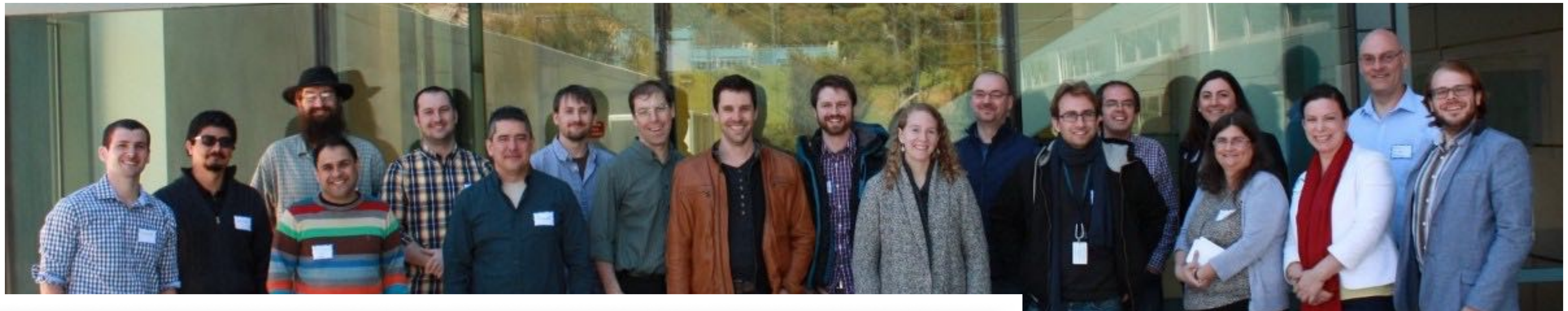
Inspiration: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

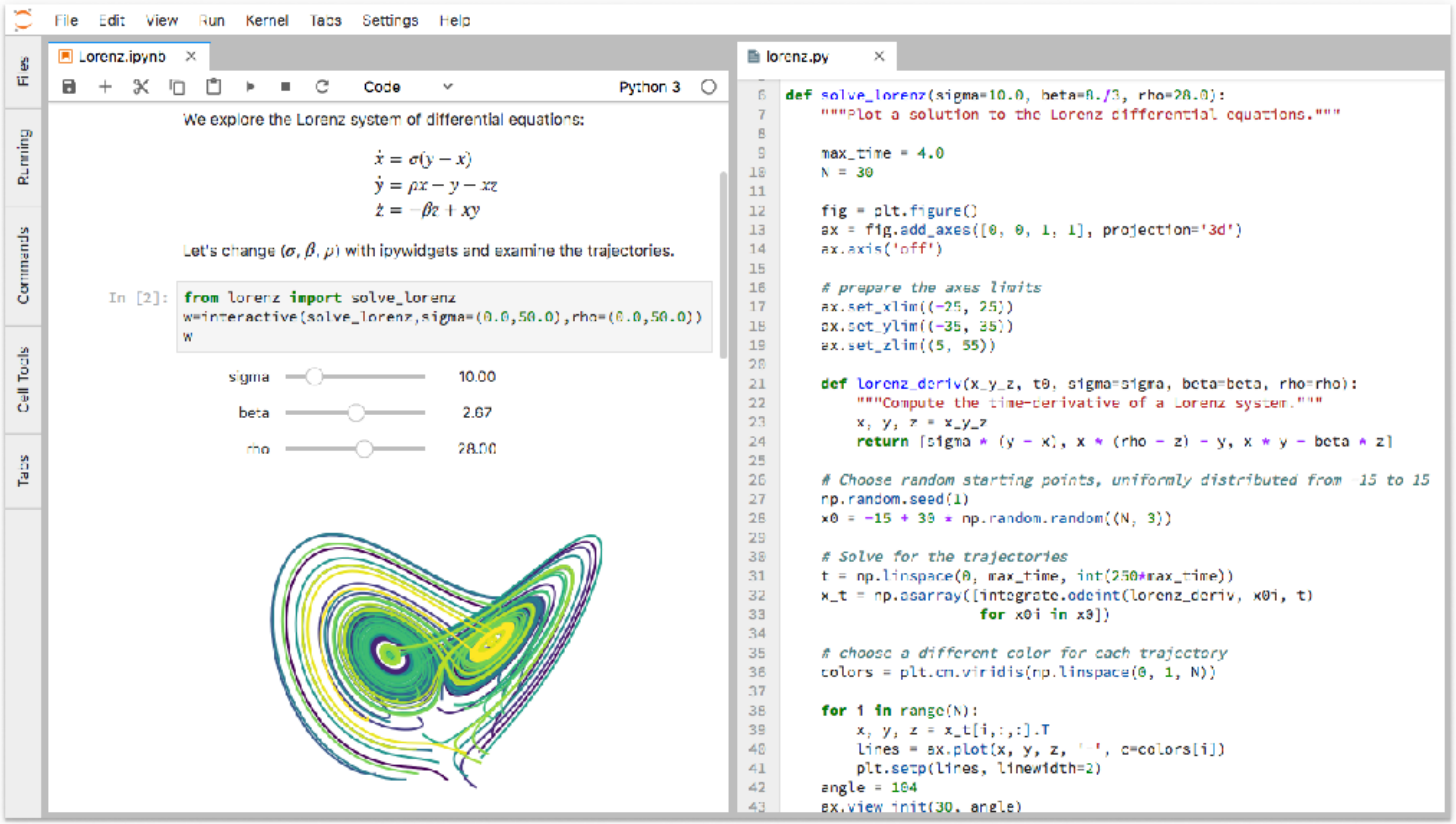
PANGEO SOFTWARE ECOSYSTEM



Inspiration: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

JUPYTER





We explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

```
In [2]: from lorenz import solve_lorenz
        w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
        w
```

sigma 10.00

beta 2.67

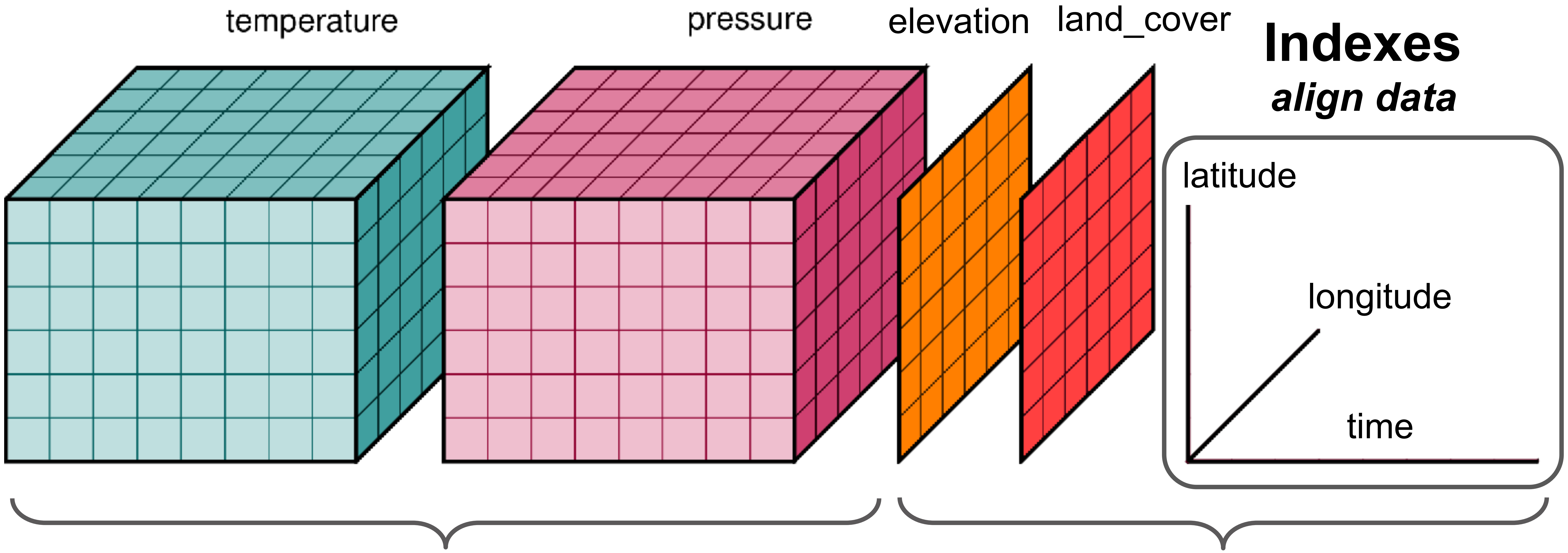
rho 28.00

```
def solve_lorenz(sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""

    max_time = 4.0
    N = 30

    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

    # prepare the axes limits
    ax.set_xlim((-25, 25))
    ax.set_ylim((-35, 35))
    ax.set_zlim((5, 55))

    def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
        """Compute the time-derivative of a Lorenz system."""
        x, y, z = x_y_z
        return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

    # Choose random starting points, uniformly distributed from -15 to 15
    np.random.seed(1)
    x0 = -15 + 30 * np.random.random((N, 3))

    # Solve for the trajectories
    t = np.linspace(0, max_time, int(250*max_time))
    x_t = np.asarray([integrate.odeint(lorenz_deriv, x0i, t)
                      for x0i in x0])

    # choose a different color for each trajectory
    colors = plt.cm.viridis(np.linspace(0, 1, N))

    for i in range(N):
        x, y, z = x_t[i,:,:].T
        lines = ax.plot(x, y, z, '-', c=colors[i])
        plt.setp(lines, linewidth=2)
    angle = 104
    ax.view_init(30, angle)
```

"Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages."

XARRAY

https://github.com/pydata/xarray



Indexes
align data

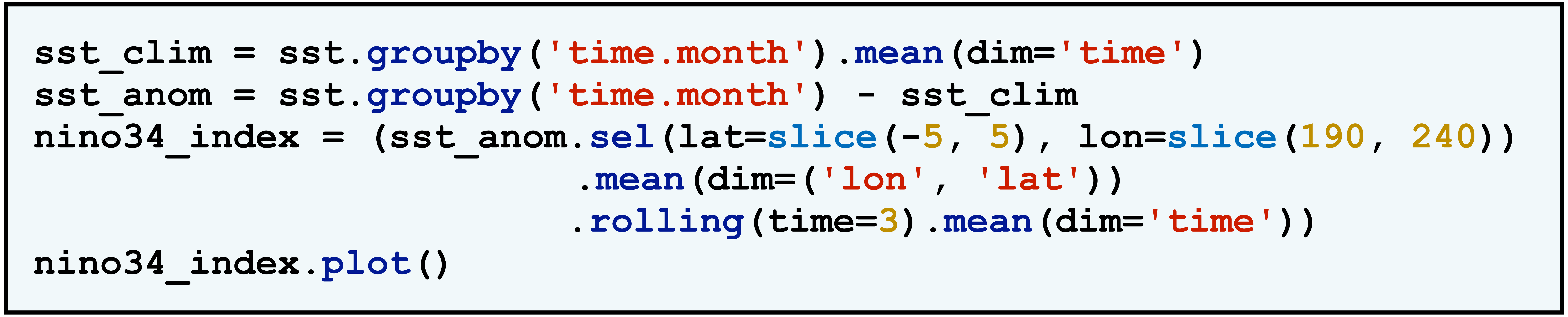
+ Attributes
metadata ignored by operations

Data variables
used for computation

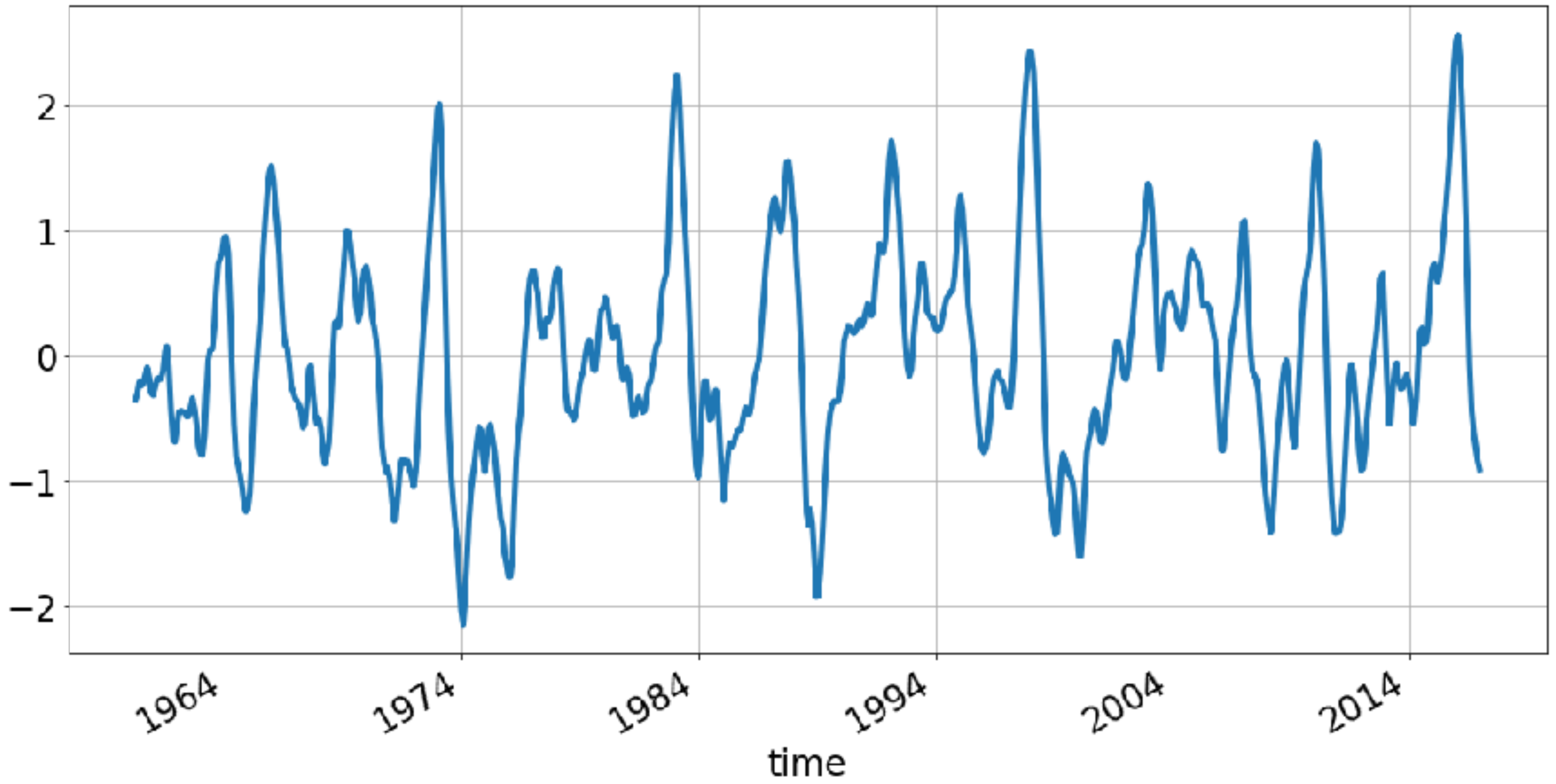
Coordinates
describe data

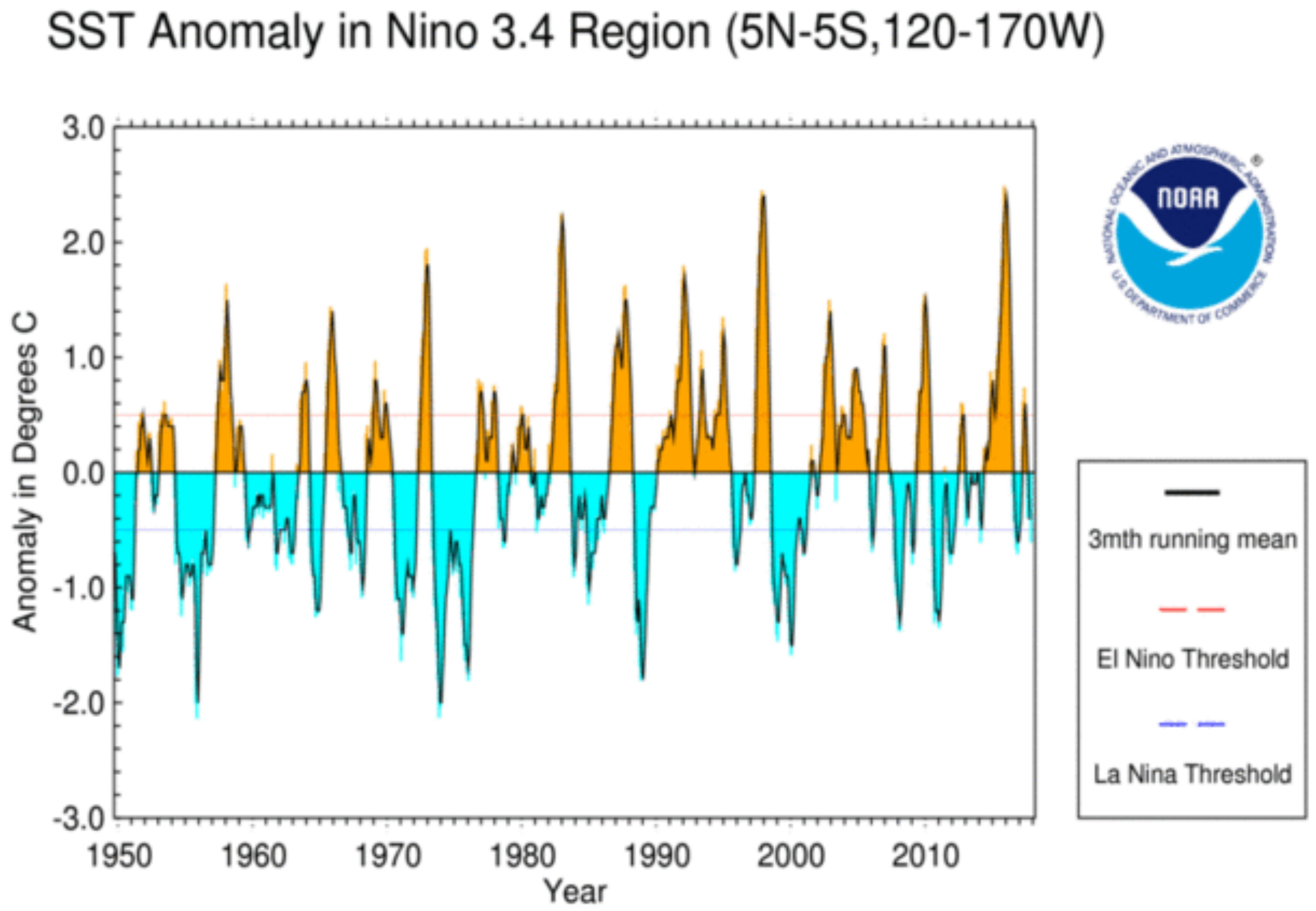
“netCDF meets pandas.DataFrame”

XARRAY: EXPRESSIVE & HIGH-LEVEL



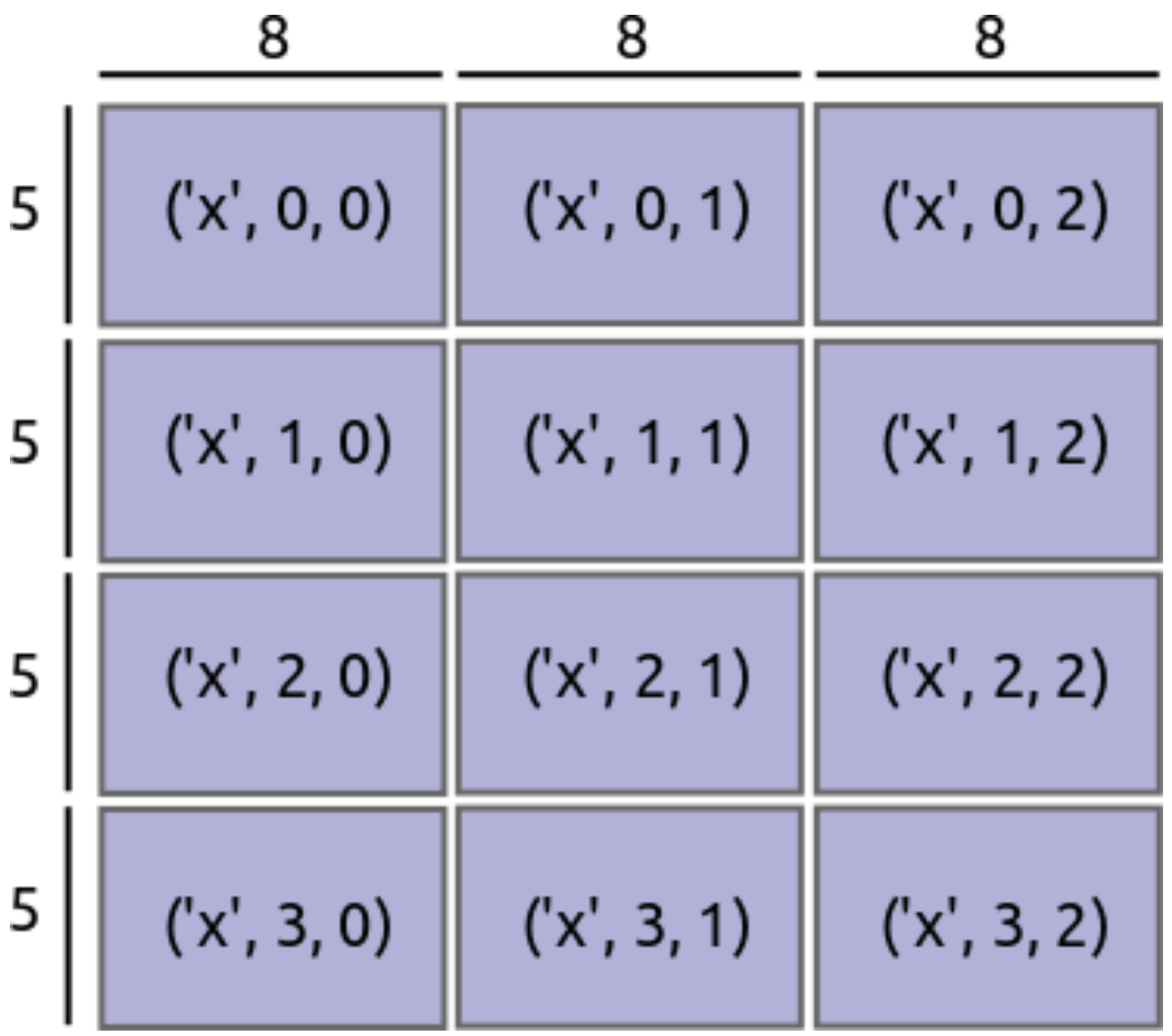
```
sst_clim = sst.groupby('time.month').mean(dim='time')
sst_anom = sst.groupby('time.month') - sst_clim
nino34_index = (sst_anom.sel(lat=slice(-5, 5), lon=slice(190, 240))
                        .mean(dim=('lon', 'lat'))
                        .rolling(time=3).mean(dim='time'))
nino34_index.plot()
```





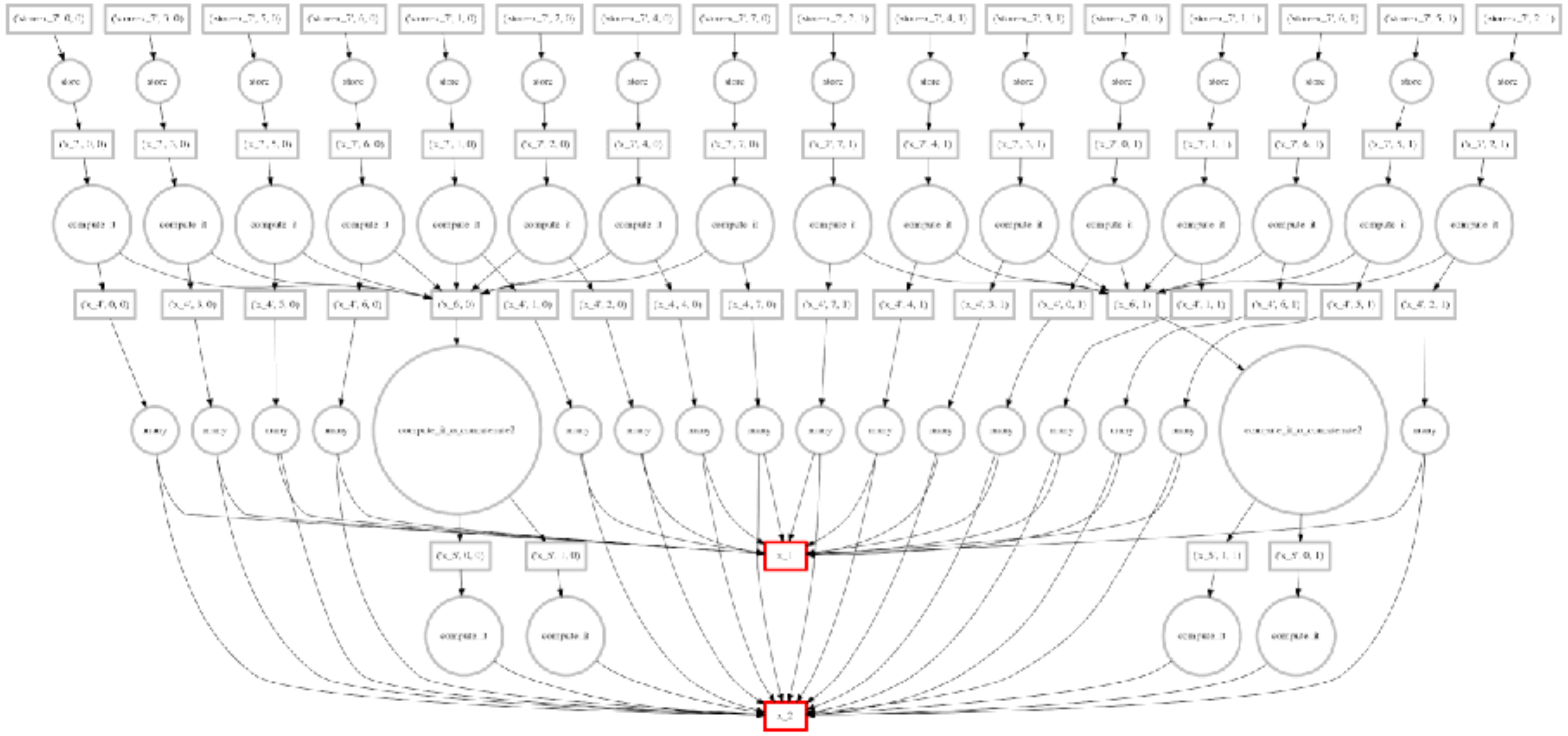
https://github.com/dask/dask/

Flexible, general-purpose parallel computing framework.



	8	8	8
5	('x', 0, 0)	('x', 0, 1)	('x', 0, 2)
5	('x', 1, 0)	('x', 1, 1)	('x', 1, 2)
5	('x', 2, 0)	('x', 2, 1)	('x', 2, 2)
5	('x', 3, 0)	('x', 3, 1)	('x', 3, 2)

ND-Arrays are split into chunks that comfortably fit in memory



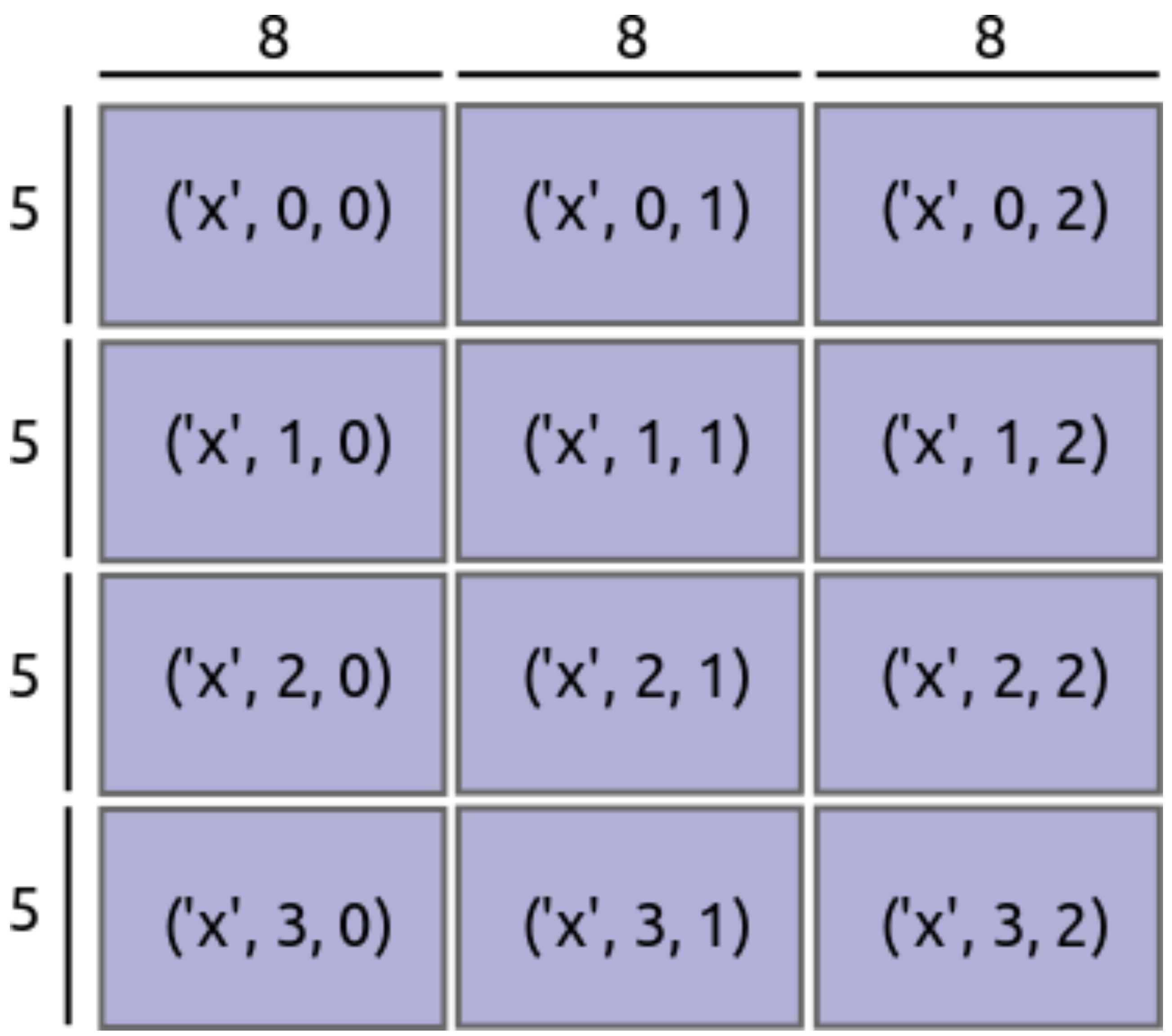
Complex computations represented as a graph of individual tasks.

Scheduler optimizes execution of graph.

DASK

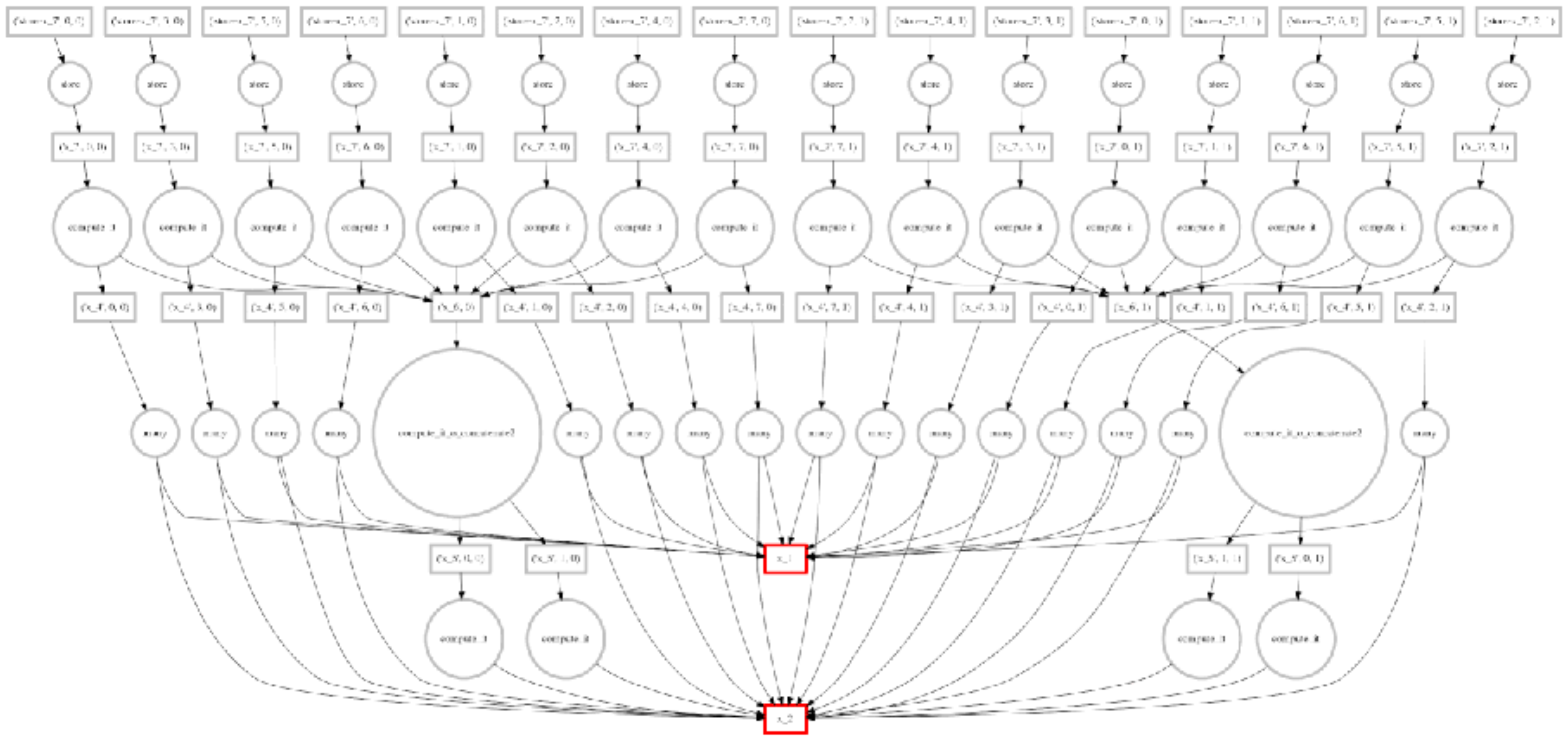
https://github.com/dask/dask/

Flexible, general-purpose parallel computing framework.



	8	8	8
5	('x', 0, 0)	('x', 0, 1)	('x', 0, 2)
5	('x', 1, 0)	('x', 1, 1)	('x', 1, 2)
5	('x', 2, 0)	('x', 2, 1)	('x', 2, 2)
5	('x', 3, 0)	('x', 3, 1)	('x', 3, 2)

ND-Arrays are split into chunks that comfortably fit in memory

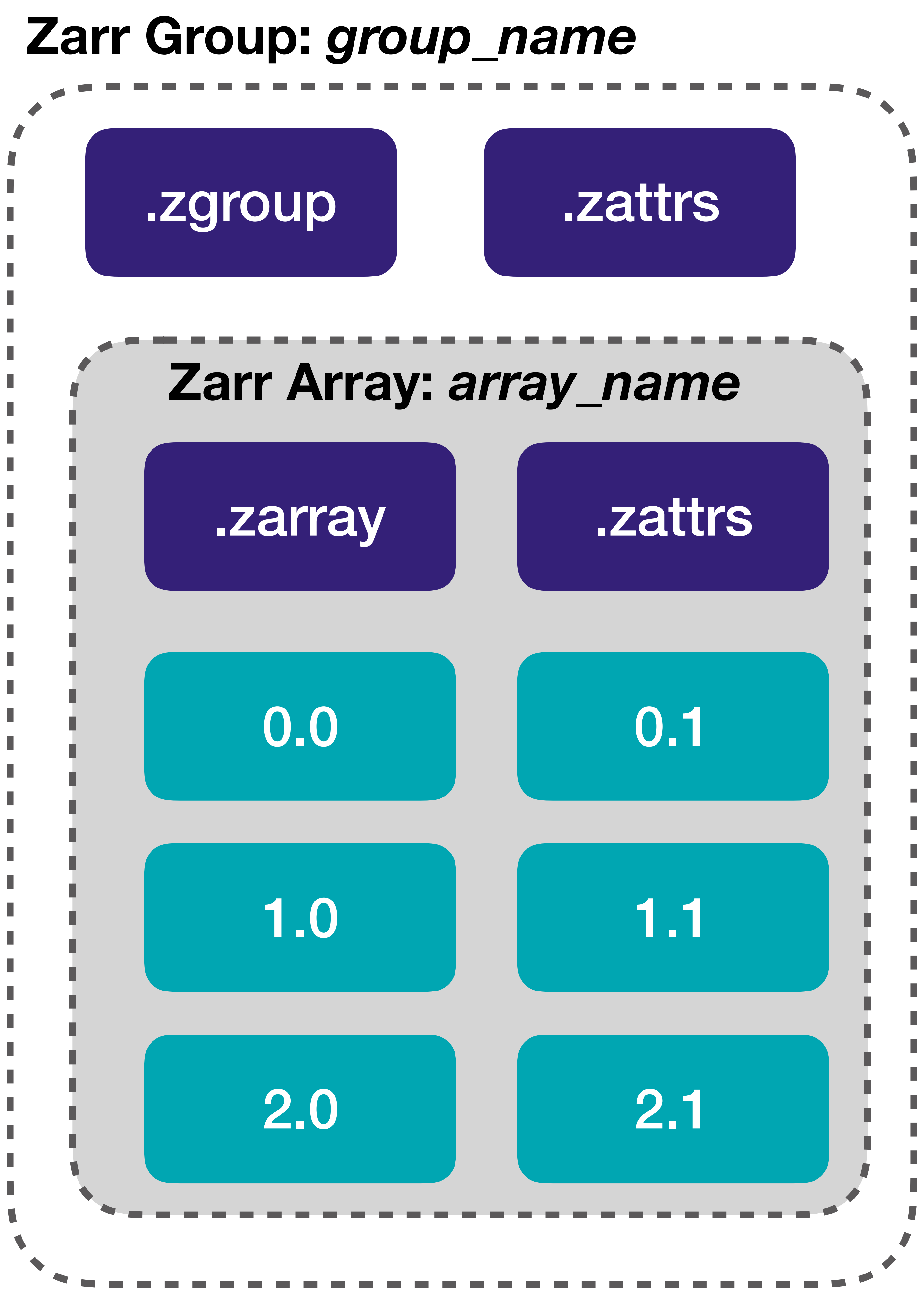


Complex computations represented as a graph of individual tasks.

Scheduler optimizes execution of graph.

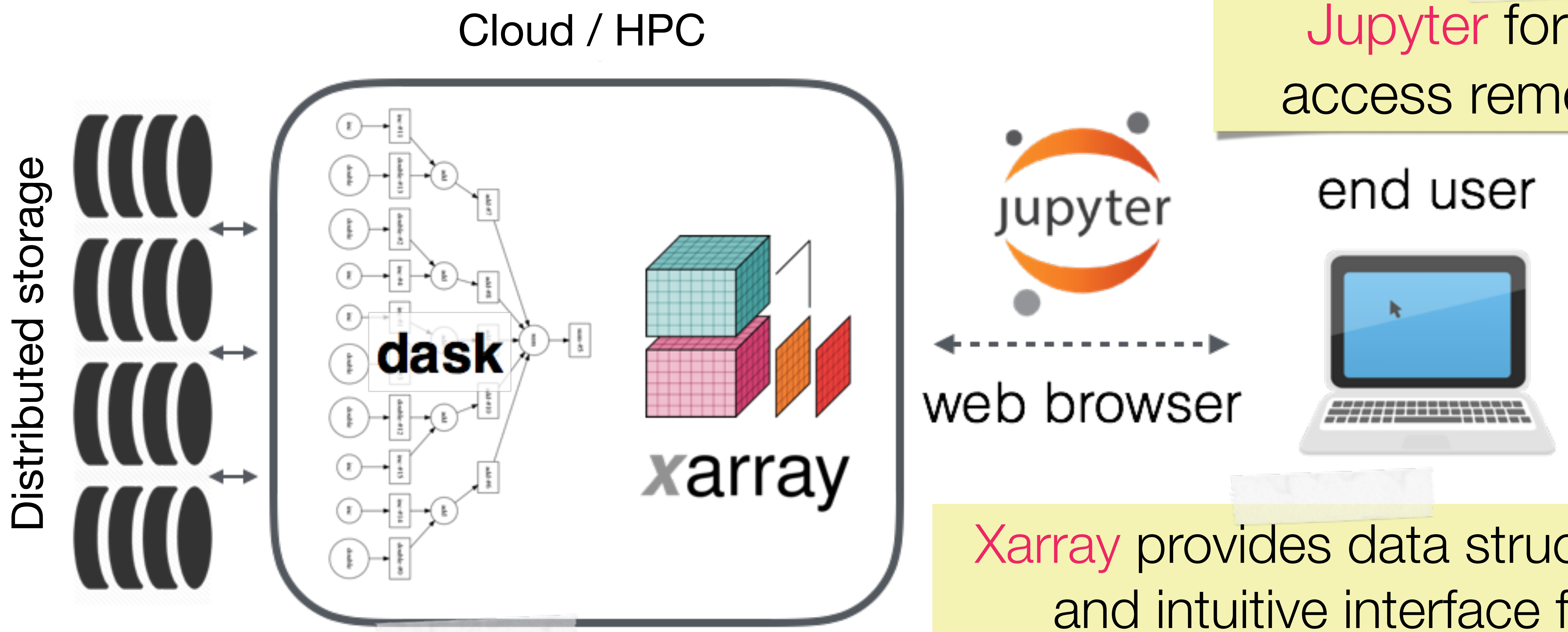
ZARR

https://zarr.readthedocs.io/



- Open source library for storage of chunked, compressed ND-arrays
- Created by Alistair Miles (Imperial) for genomics research (*@alimanfoo*); now community supported standard
- Arrays are split into user-defined chunks; each chunk is optional compressed (zlib, zstd, etc.)
- Can store arrays in memory, directories, zip files, or any python mutable mapping interface (dictionary)
- External libraries (s3fs, gcsf) provide a way to store directly into cloud object storage
- Implementations in Python, C++, Java (N5), Julia, Javascript

PANGEO ARCHITECTURE



Jupyter for interactive access remote systems

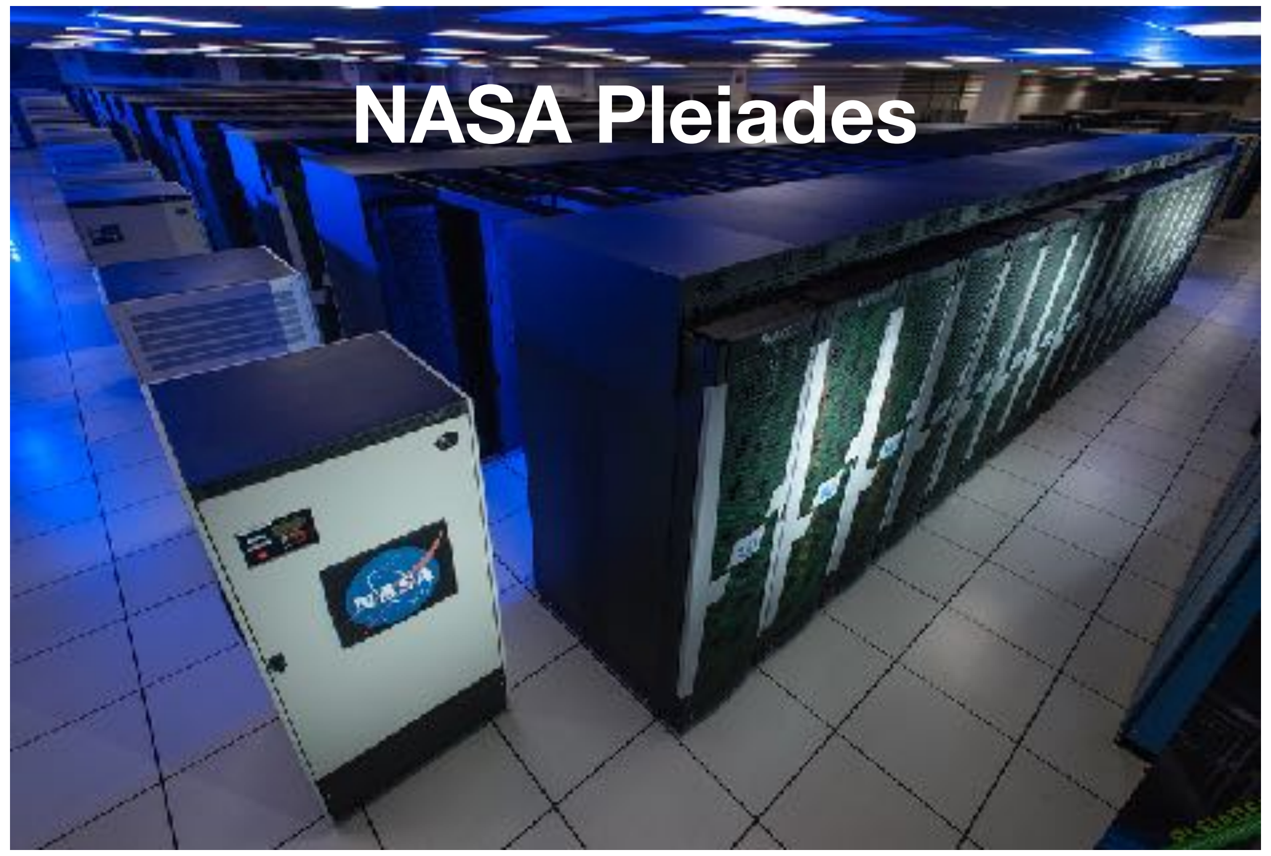
"Analysis Ready Data" stored on globally-available distributed storage.

Xarray provides data structures and intuitive interface for interacting with datasets

Parallel computing system allows users deploy clusters of compute nodes for data processing.

Dask tells the nodes what to do.

PANGEO DEPLOYMENTS


NASA Pleiades

NCAR Cheyenne

OCEAN.PANGEO.IO

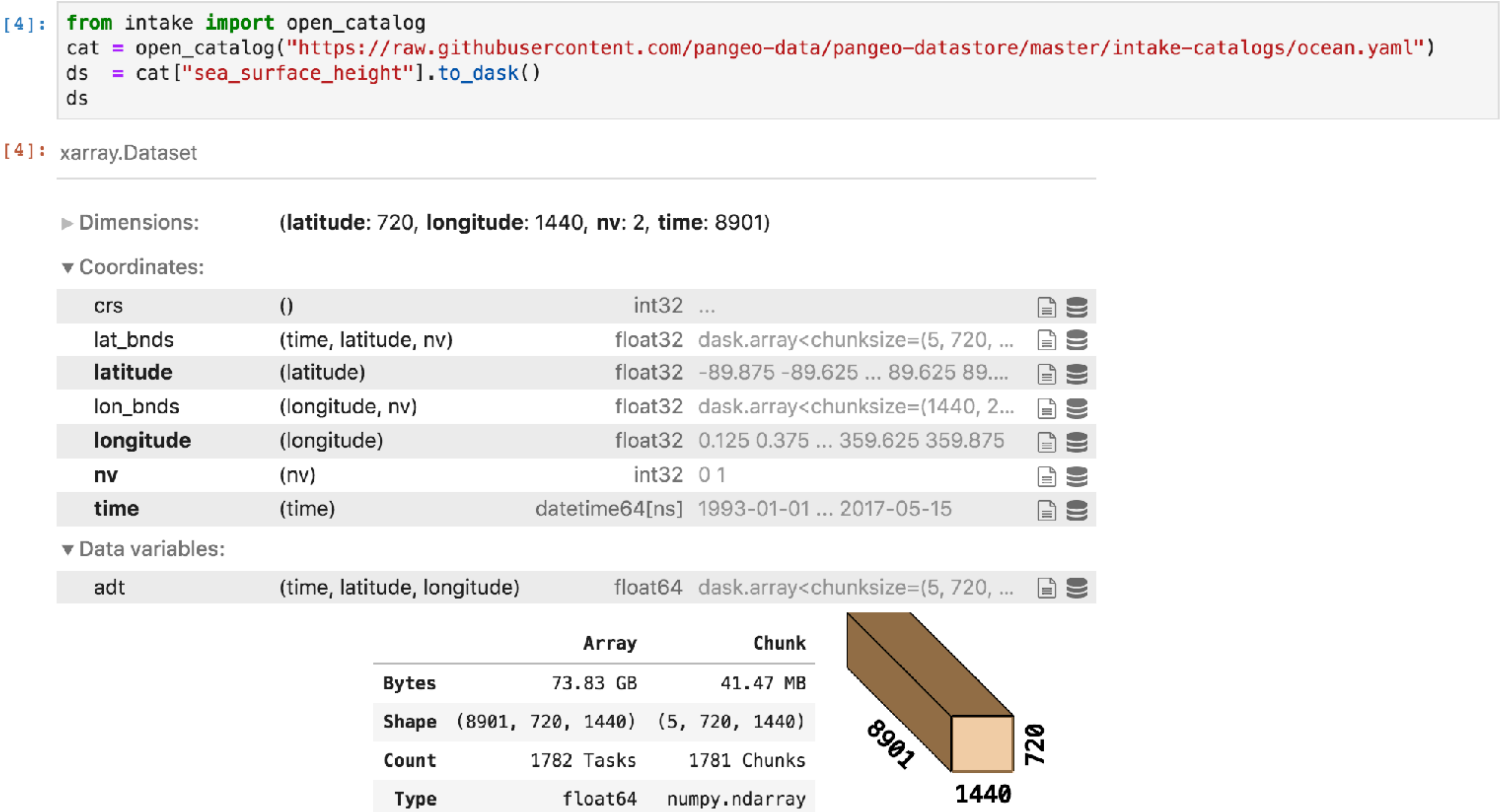
Google Cloud Platform

Microsoft Azure

aws

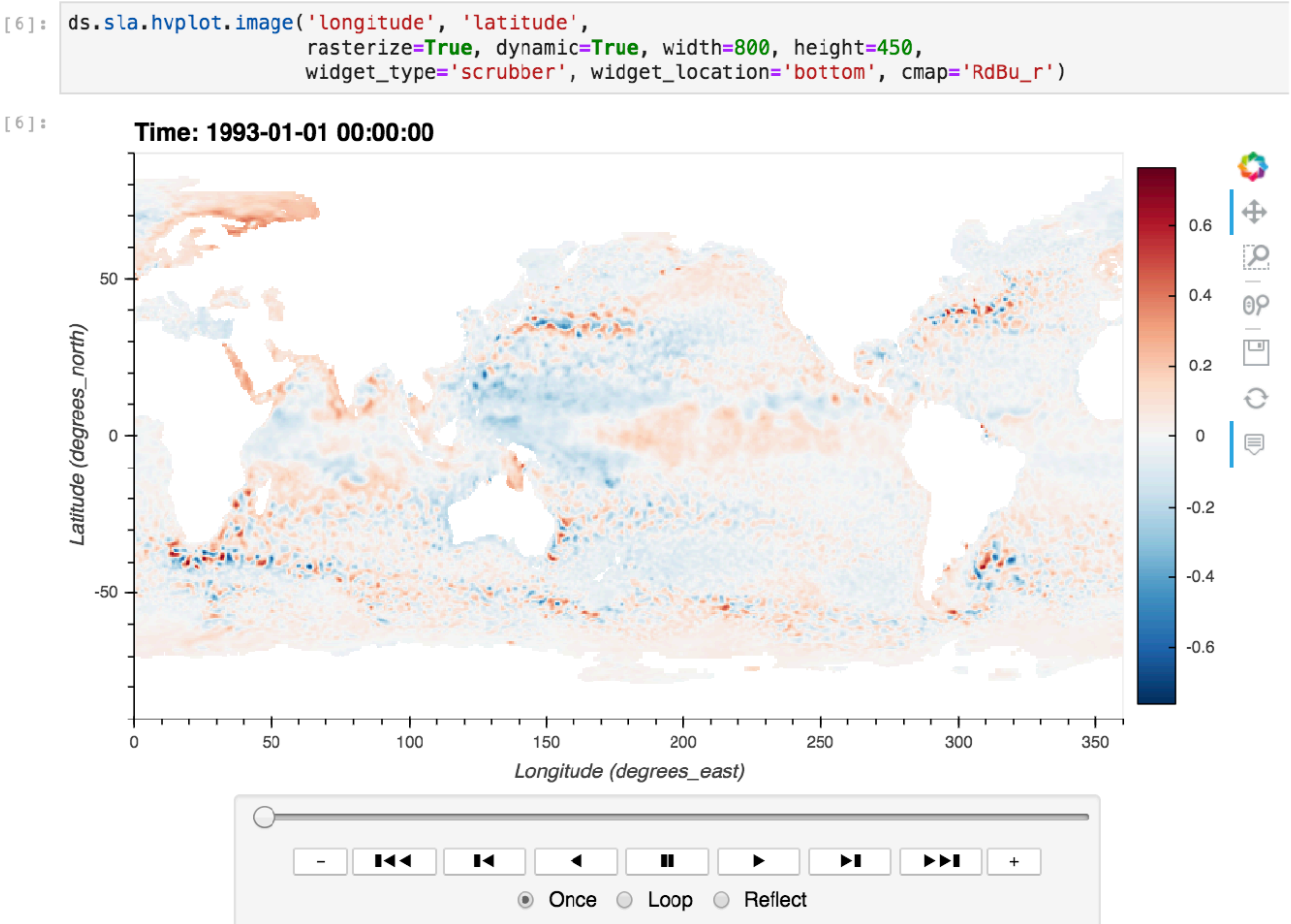
HTTP://PANGEO.IO/DEPLOYMENTS.HTML

Live example at
http://gallery.pangeo.io/repos/pangeo-gallery/physical-oceanography/01_sea-surface-height.html



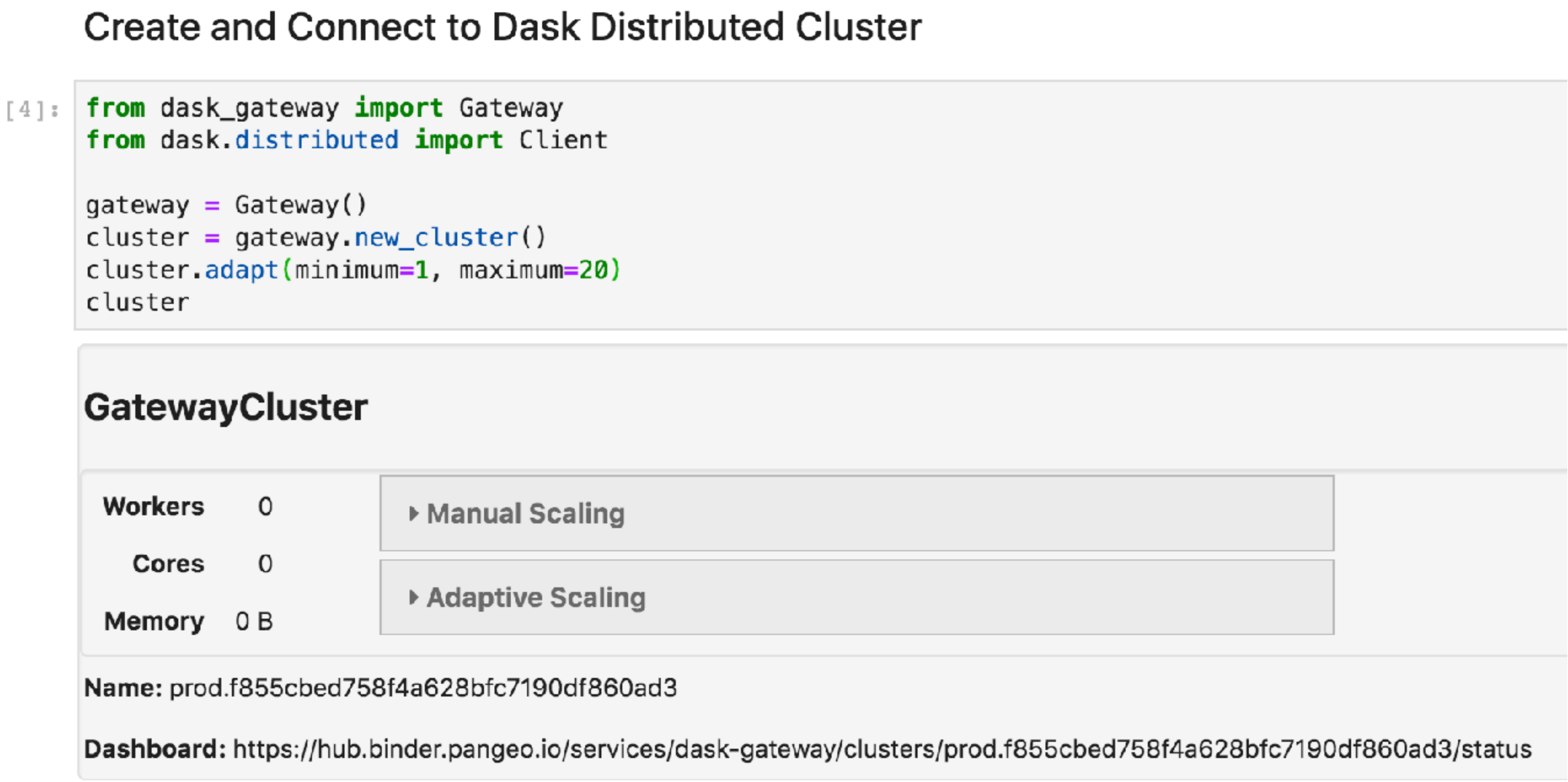
```
[4]: from intake import open_catalog
     cat = open_catalog("https://raw.githubusercontent.com/pangeo-data/pangeo-datastore/master/intake-catalogs/ocean.yaml")
     ds  = cat["sea_surface_height"].to_dask()
     ds
```

```
[4]: xarray.Dataset

  ▶ Dimensions:      (latitude: 720, longitude: 1440, nv: 2, time: 8901)

  ▼ Coordinates:
      crs            ()                                  int32  ...
      lat_bnds       (time, latitude, nv)              float32  dask.array<chunksize=(5, 720, ...
      latitude       (latitude)                        float32  -89.875 -89.625 ... 89.625 89....
      lon_bnds       (longitude, nv)                   float32  dask.array<chunksize=(1440, 2...
      longitude      (longitude)                       float32  0.125 0.375 ... 359.625 359.875
      nv             (nv)                                int32  0 1
      time           (time)                     datetime64[ns]  1993-01-01 ... 2017-05-15

  ▼ Data variables:
      adt            (time, latitude, longitude)       float64  dask.array<chunksize=(5, 720, ...

                              Array                 Chunk
          Bytes            73.83 GB              41.47 MB
          Shape   (8901, 720, 1440)      (5, 720, 1440)
          Count          1782 Tasks           1781 Chunks
           Type             float64         numpy.ndarray
```



```
[6]: ds.sla.hvplot.image('longitude', 'latitude',
                         rasterize=True, dynamic=True, width=800, height=450,
                         widget_type='scrubber', widget_location='bottom', cmap='RdBu_r')
```

[6]:



Create and Connect to Dask Distributed Cluster

[4]:

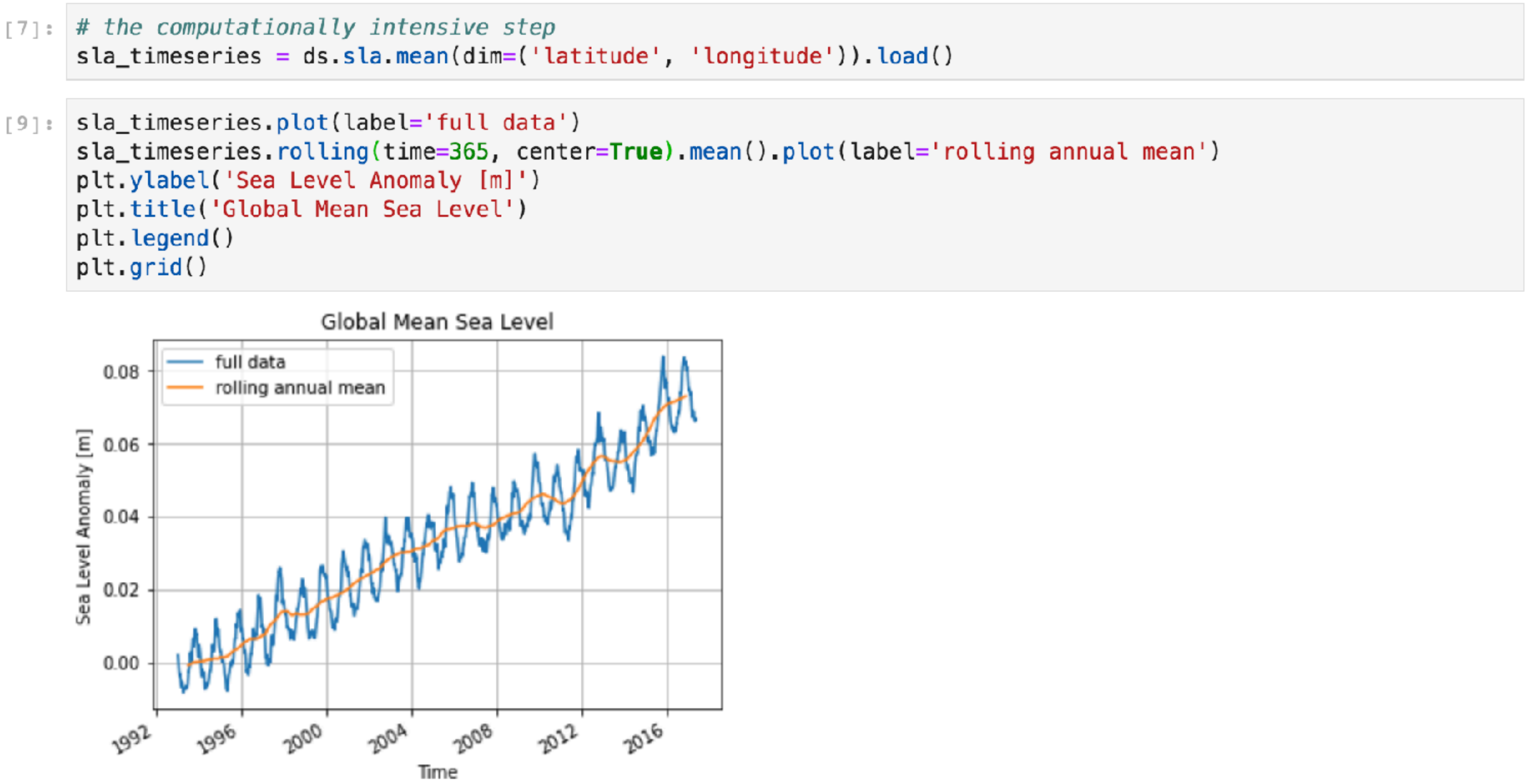
```python
from dask_gateway import Gateway
from dask.distributed import Client

gateway = Gateway()
cluster = gateway.new_cluster()
cluster.adapt(minimum=1, maximum=20)
cluster
```

GatewayCluster

Workers	0
Cores	0
Memory	0 B

Name: prod.f855cbed758f4a628bfc7190df860ad3

Dashboard: https://hub.binder.pangeo.io/services/dask-gateway/clusters/prod.f855cbed758f4a628bfc7190df860ad3/status

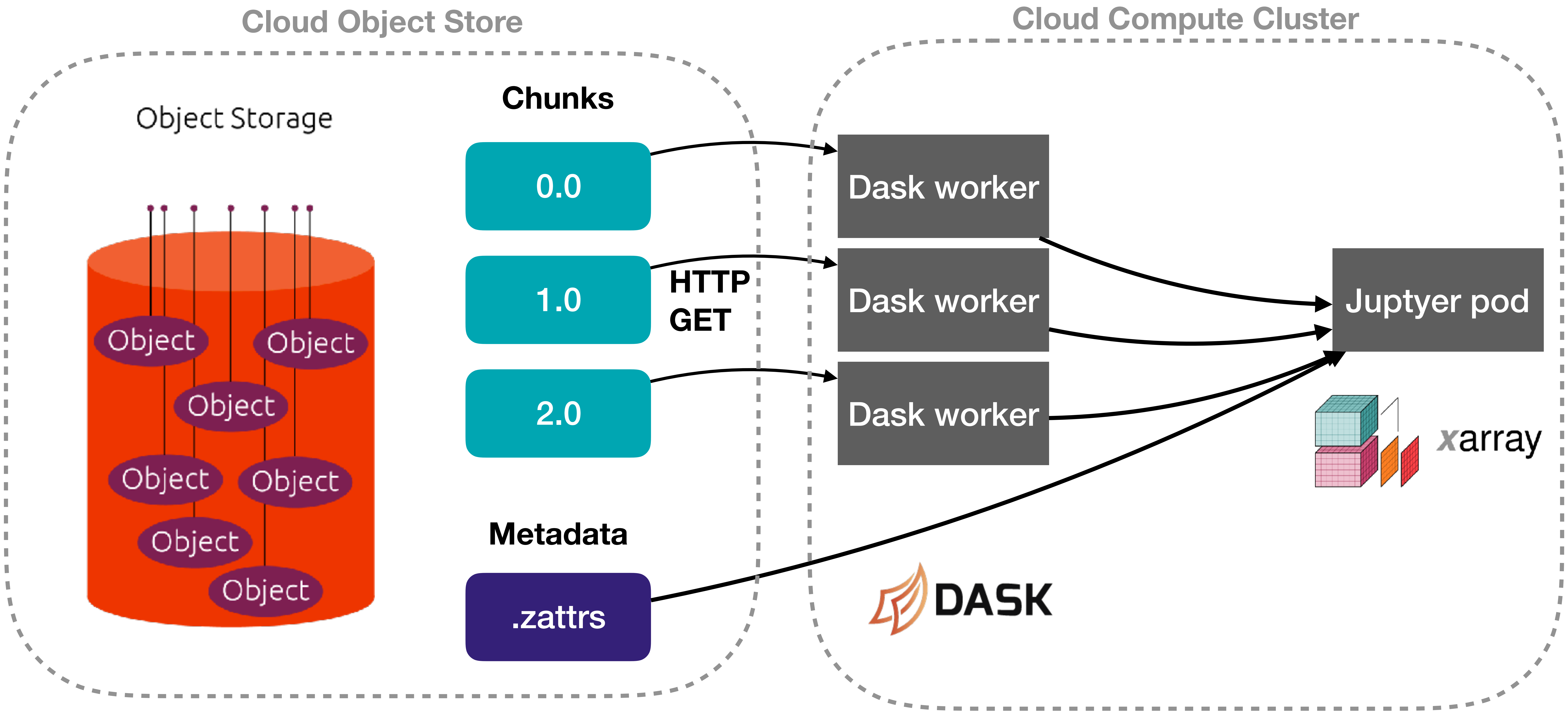


```
[7]: # the computationally intensive step
     sla_timeseries = ds.sla.mean(dim=('latitude', 'longitude')).load()
```

```
[9]: sla_timeseries.plot(label='full data')
     sla_timeseries.rolling(time=365, center=True).mean().plot(label='rolling annual mean')
     plt.ylabel('Sea Level Anomaly [m]')
     plt.title('Global Mean Sea Level')
     plt.legend()
     plt.grid()
```

SHARING DATA IN THE CLOUD ERA

Pangeo Approach: Direct Access to Cloud Object Storage



Do we need an API for data access?

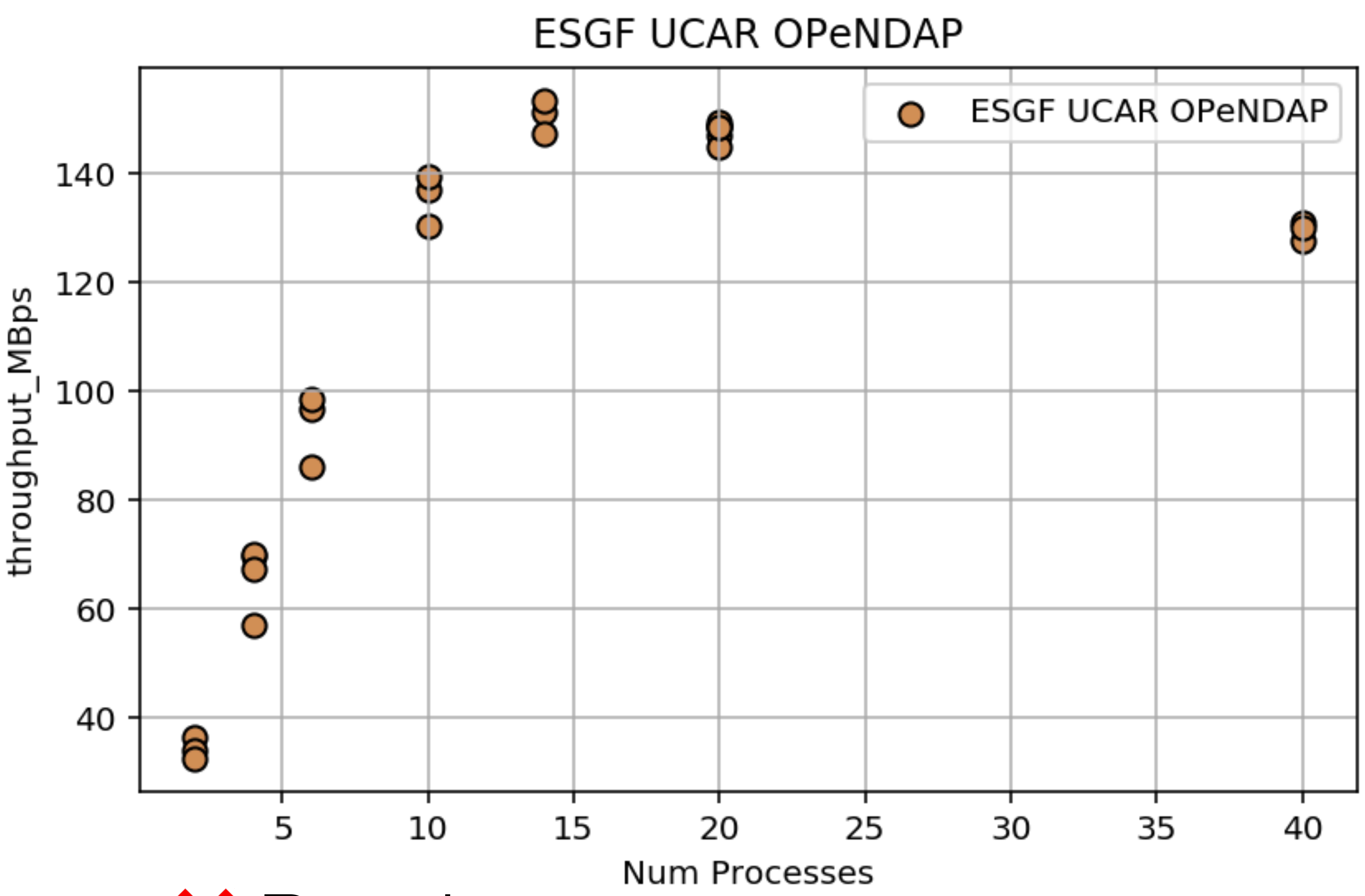
OPENDAP

- ✅ Access remote netCDF-style datasets over HTTP
- ✅ Subset based on coordinates / variables
- ✅ Load data lazily
- ❌ Requires a server

XARRAY + ZARR

- ✅ Access remote netCDF-style datasets over HTTP
- ✅ Subset based on coordinates / variables
- ✅ Load data lazily
- ✅ Serverless (only uses S3)

OPENDAP



❌ Requires a server

XARRAY + ZARR

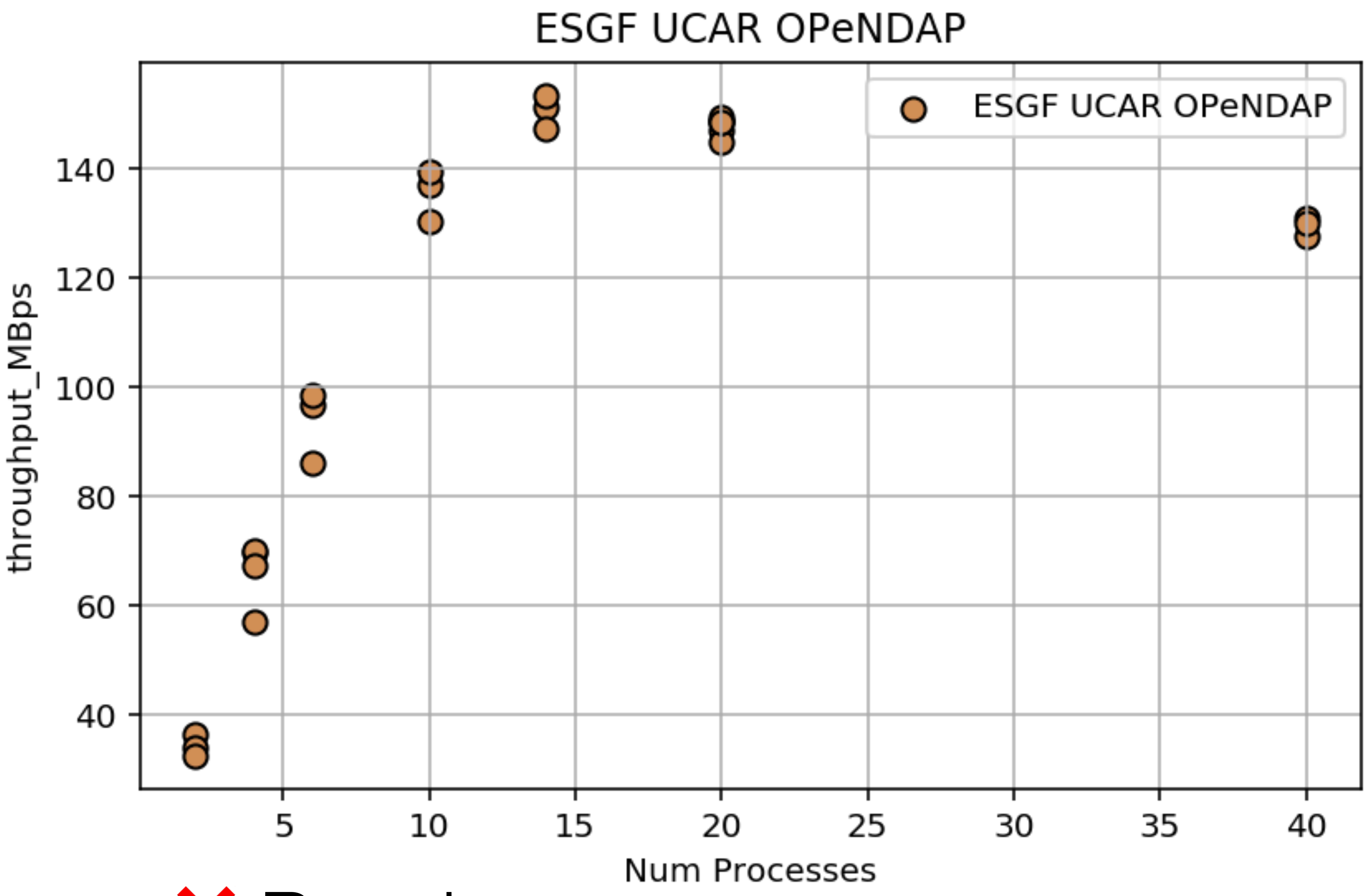
✅ Access remote netCDF-style datasets over HTTP

✅ Subset based on coordinates / variables

✅ Load data lazily

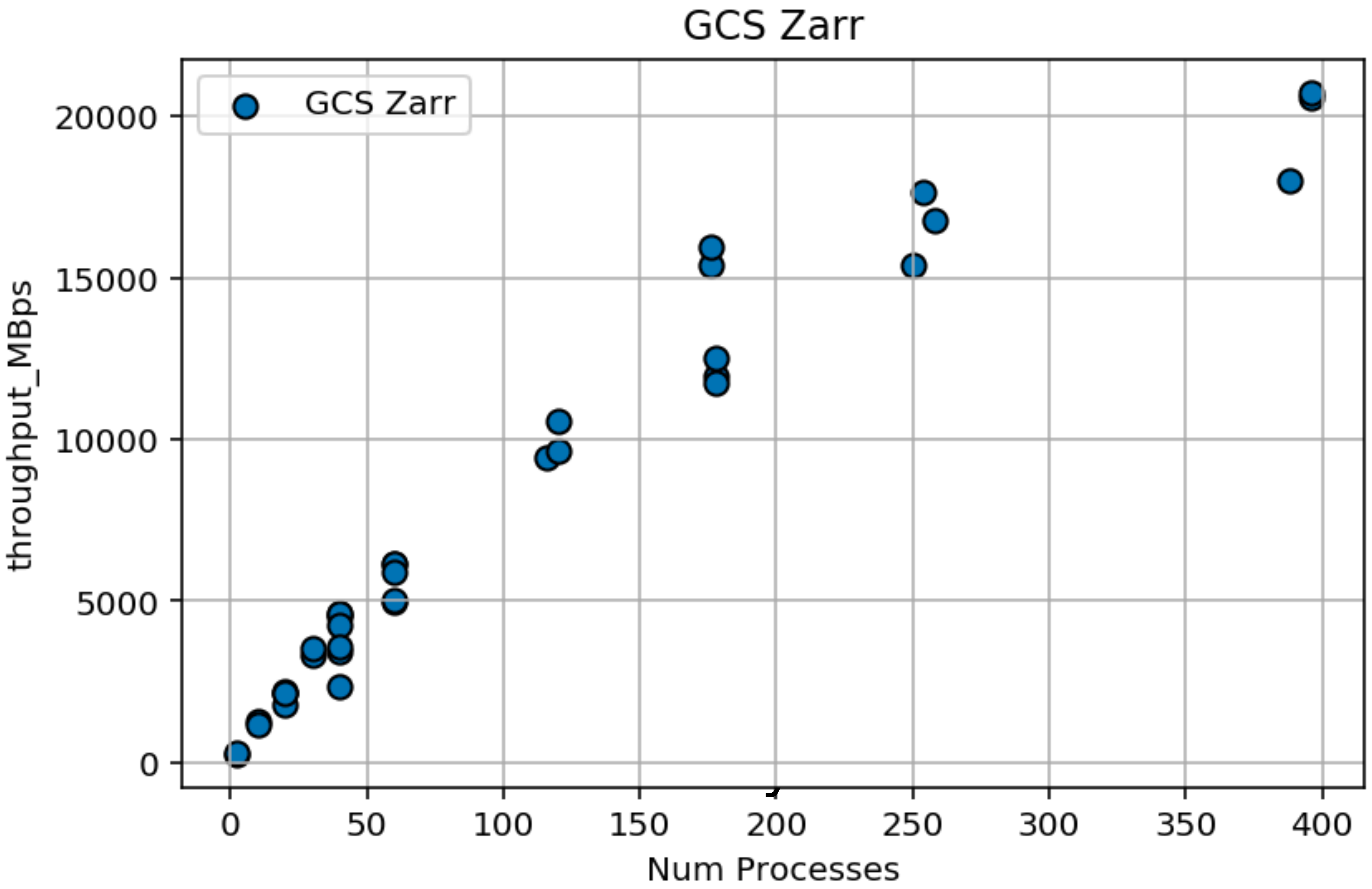
✅ Serverless (only uses S3)

OPENDAP



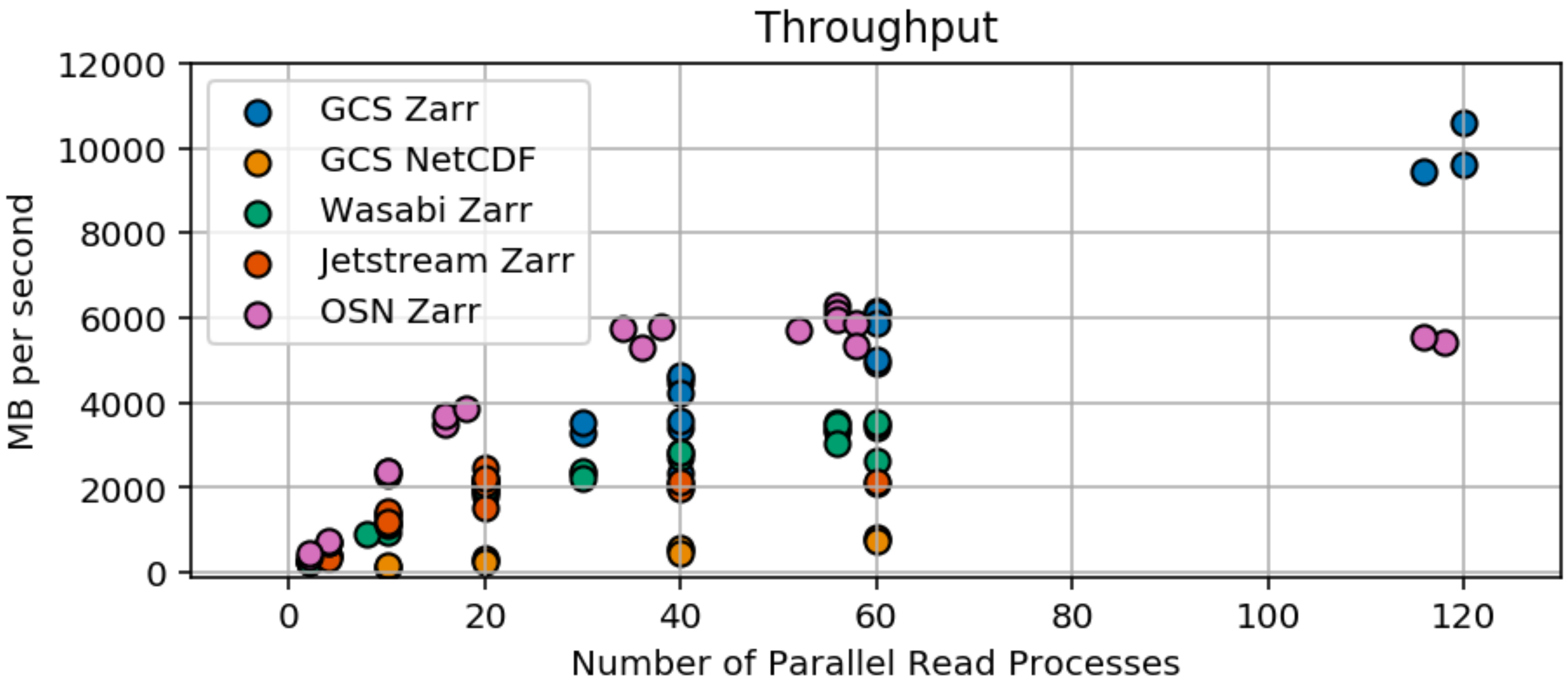
❌ Requires a server

XARRAY + ZARR



✅ Serverless (only uses S3)

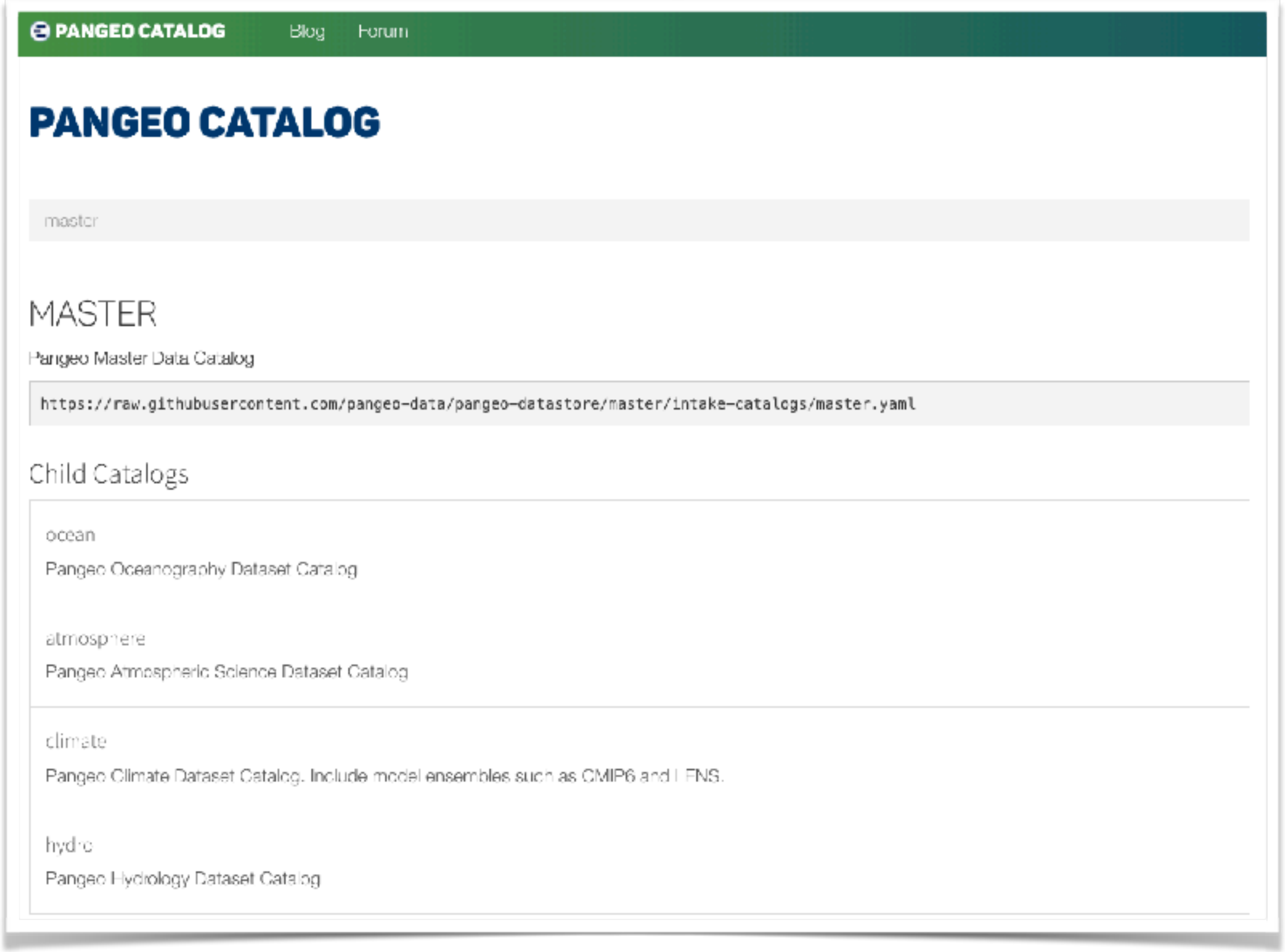
CLOUD STORAGE THROUGHPUT BENCHMARKS



To Google Cloud Region US-CENTRAL

PANGEO CLOUD DATA CATALOG

CATALOG.PANGEO.IO



PANGEO CATALOG

master

MASTER

Pangeo Master Data Catalog

```
https://raw.githubusercontent.com/pangeo-data/pangeo-datastore/master/intake-catalogs/master.yaml
```

Child Catalogs

ocean

Pangeo Oceanography Dataset Catalog

atmosphere

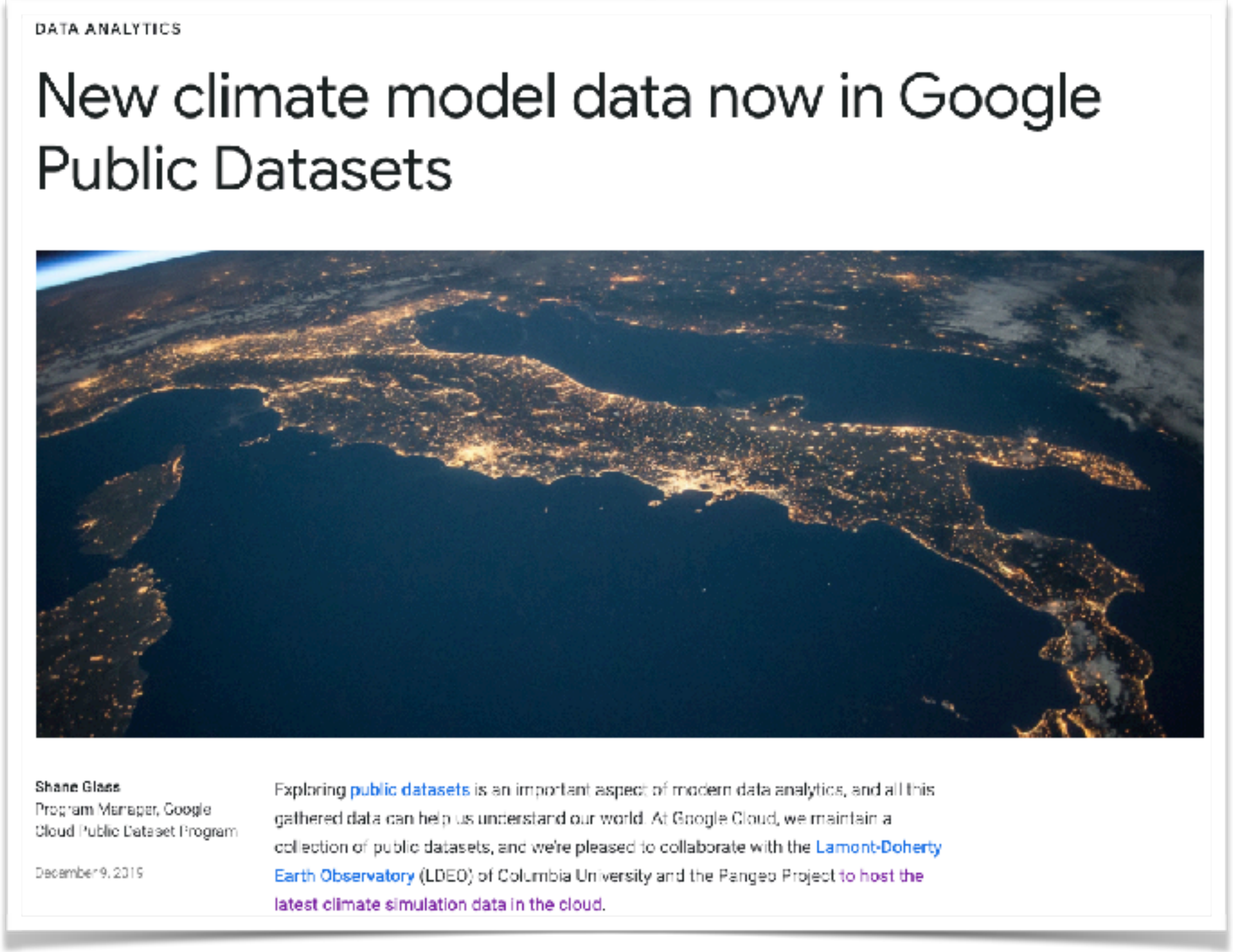
Pangeo Atmospheric Science Dataset Catalog

climate

Pangeo Climate Dataset Catalog. Include model ensembles such as CMIP6 and LENS.

hydro

Pangeo Hydrology Dataset Catalog



DATA ANALYTICS

New climate model data now in Google Public Datasets

Shane Glass
Program Manager, Google Cloud Public Dataset Program

December 9, 2019

Exploring public datasets is an important aspect of modern data analytics, and all this gathered data can help us understand our world. At Google Cloud, we maintain a collection of public datasets, and we're pleased to collaborate with the Lamont-Doherty Earth Observatory (LDEO) of Columbia University and the Pangeo Project to host the latest climate simulation data in the cloud.

TAKE-HOME MESSAGE TO WGISS

- Open source scientific python provides a great foundation for scalable earth system analytics (especially in the cloud). ***Use it, don't reinvent it!***

- How do we support / sustain open-source foundational software tools? (No agency or lab "owns" these, but they are critical infrastructure.)

  idea: pay your staff to contribute to Xarray, Dask, etc., don't just build on top of them

- The best way to take advantage of cloud is to ***give users direct access to analysis-ready data in object storage***. Don't hide it behind an API.

LEARN MORE

http://pangeo.io

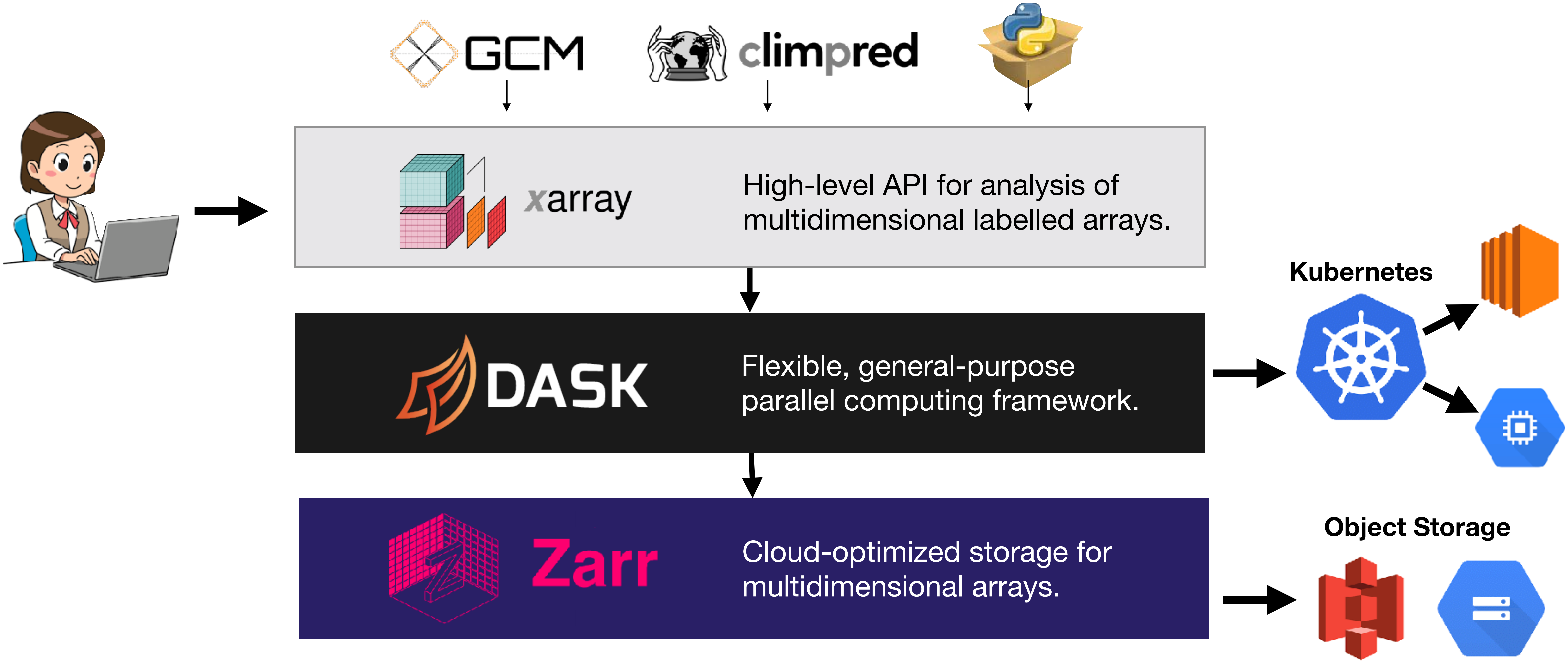
https://github.com/pangeo-data/

https://medium.com/pangeo

@pangeo_data

Extra Slides

PANGEO CLOUD STACK



High-level API for analysis of multidimensional labelled arrays.

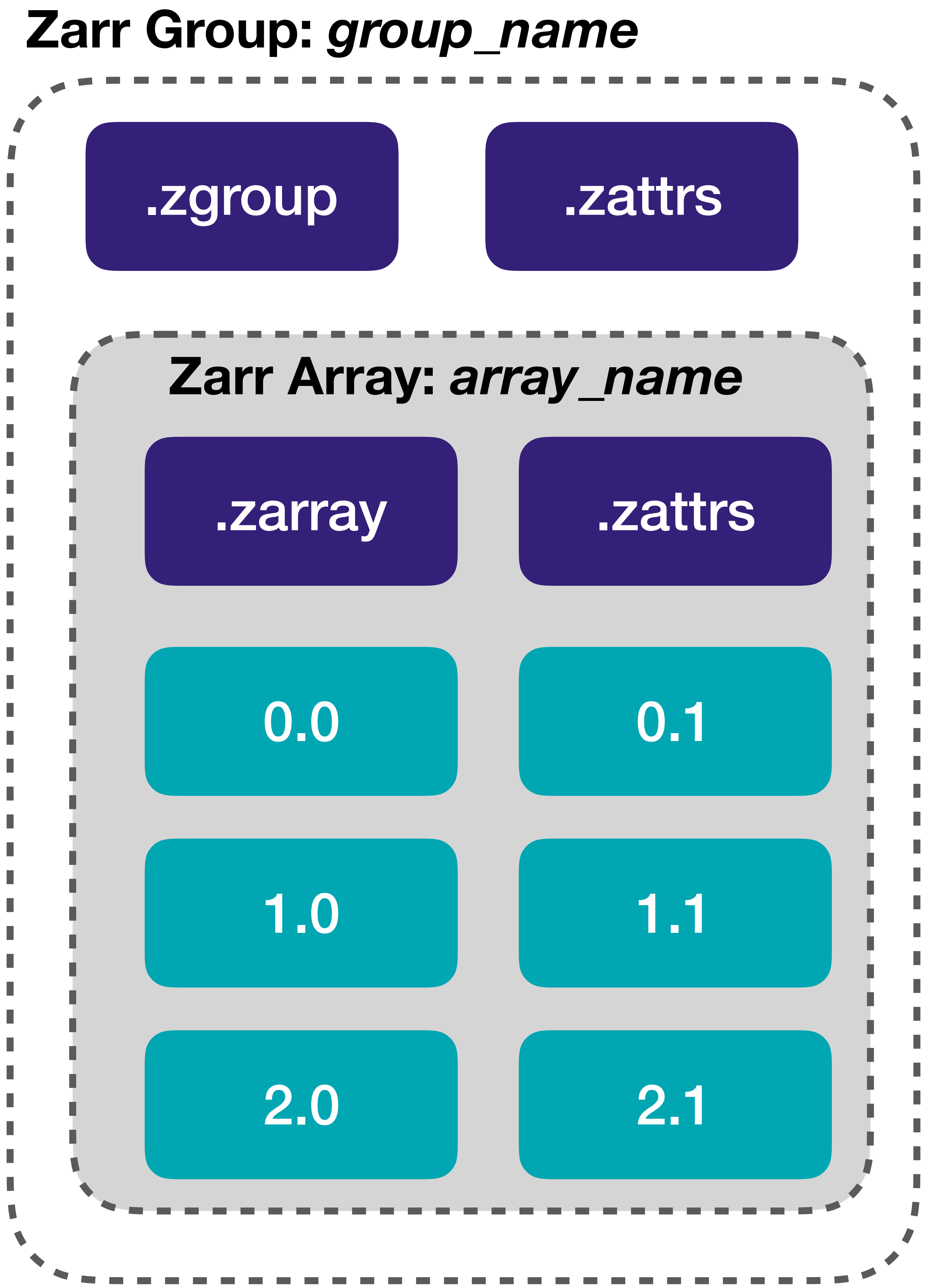
Flexible, general-purpose parallel computing framework.

Cloud-optimized storage for multidimensional arrays.

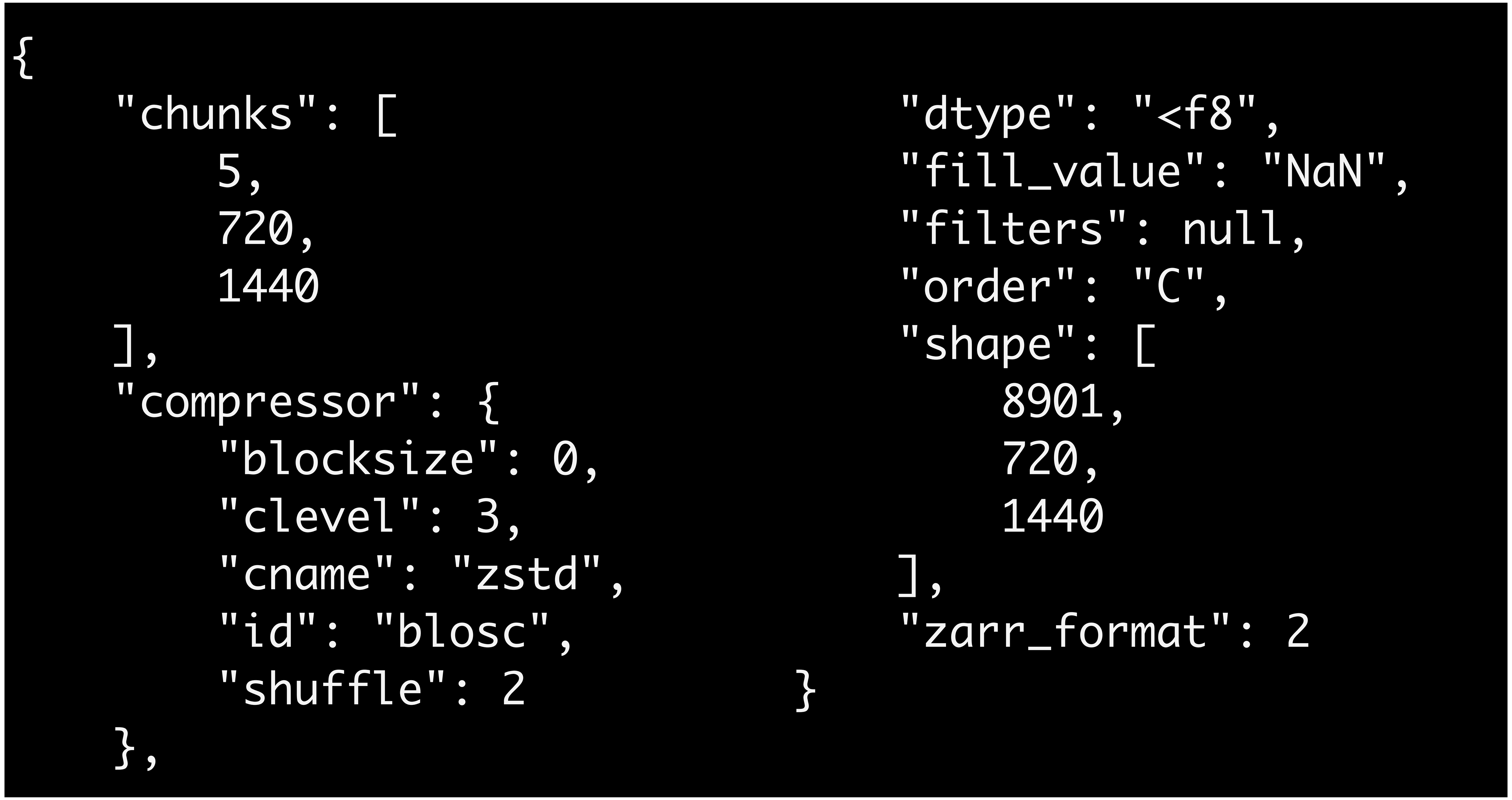
Kubernetes

Object Storage

ZARR

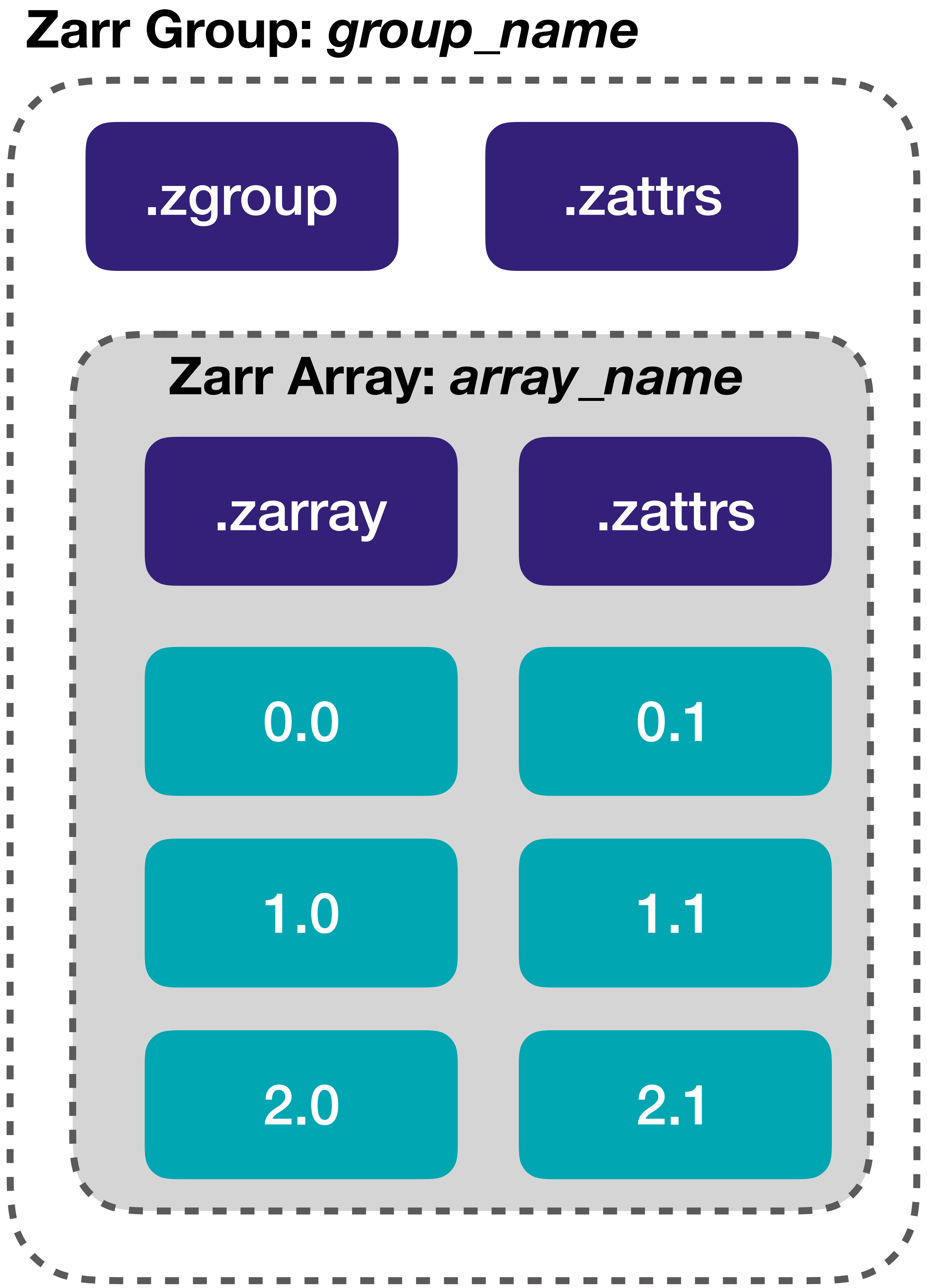


Example .zarray file (json)



```
{
    "chunks": [
        5,
        720,
        1440
    ],
    "compressor": {
        "blocksize": 0,
        "clevel": 3,
        "cname": "zstd",
        "id": "blosc",
        "shuffle": 2
    },
    "dtype": "<f8",
    "fill_value": "NaN",
    "filters": null,
    "order": "C",
    "shape": [
        8901,
        720,
        1440
    ],
    "zarr_format": 2
}
```

ZARR



Zarr Group: *group_name*

.zgroup

.zattrs

Zarr Array: *array_name*

.zarray

.zattrs

0.0

0.1

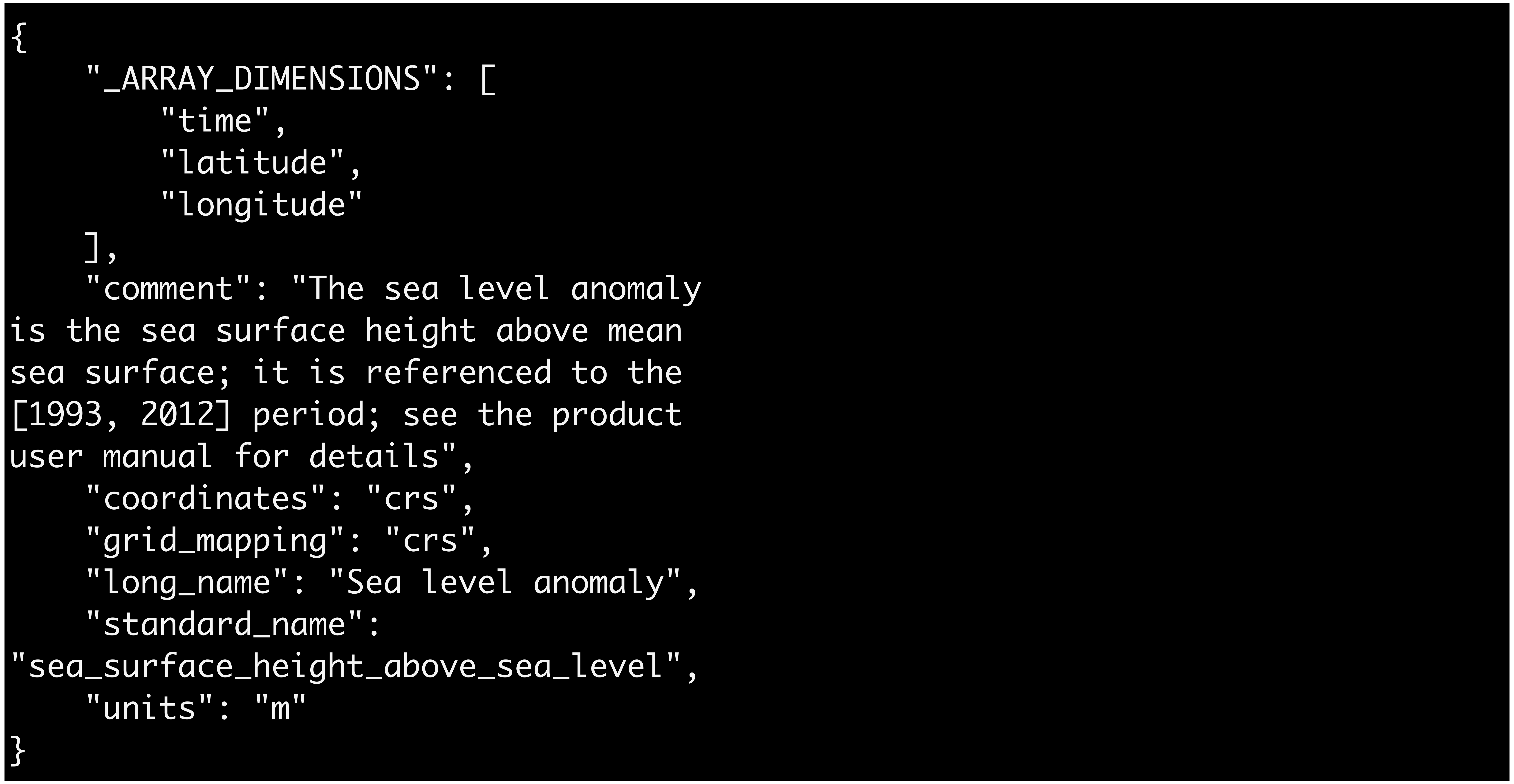
1.0

1.1

2.0

2.1

Example .attrs file (json)



```
{
    "_ARRAY_DIMENSIONS": [
        "time",
        "latitude",
        "longitude"
    ],
    "comment": "The sea level anomaly
is the sea surface height above mean
sea surface; it is referenced to the
[1993, 2012] period; see the product
user manual for details",
    "coordinates": "crs",
    "grid_mapping": "crs",
    "long_name": "Sea level anomaly",
    "standard_name":
"sea_surface_height_above_sea_level",
    "units": "m"
}
```